

[illegible]

```

EEEEEEEEEE XX      XX      CCCCCCCC DDDDDDDD BBBB BBBB GGGGGGGG
EEEEEEEEEE XX      XX      CCCCCCCC DDDDDDDD BBBB BBBB GGGGGGGG
EE          XX      XX      CC          DD      DD      BB      BB      GG
EE          XX      XX      CC          DD      DD      BB      BB      GG
EE          XX      XX      CC          DD      DD      BB      BB      GG
EEEEEEEEEE XX      XX      CC          DD      DD      BBBB BBBB GG
EEEEEEEEEE XX      XX      CC          DD      DD      BBBB BBBB GG
EE          XX      XX      CC          DD      DD      BB      BB      GG
EE          XX      XX      CC          DD      DD      BB      BB      GG
EE          XX      XX      CC          DD      DD      BB      BB      GG
EEEEEEEEEE XX      XX      CCCCCCCC DDDDDDDD BBBB BBBB GGGGGG
EEEEEEEEEE XX      XX      CCCCCCCC DDDDDDDD BBBB BBBB GGGGGG
    
```

```

LL          IIIIII SSSSSSSS
LL          IIIIII SSSSSSSS
LL          II      SS
LL          II      SS
LL          II      SS
LL          II      SS
LL          II      SSSSSS
LL          II      SSSSSS
LL          II      SS
LL          II      SS
LL          II      SS
LLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLL IIIIII SSSSSSSS
    
```

```
0001 0 MODULE  exch$dbg                                %TITLE 'Conditionally assembled debugging routines'
0002 0
0003 0 (
0004 0 IDENT = 'V04-000',
0005 0 ADDRESSING_MODE (EXTERNAL=LONG_RELATIVE, NONEXTERNAL=WORD_RELATIVE)
0006 0 ) =
0007 1 BEGIN
0008 1 *****
0009 1 *
0010 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0011 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0012 1 * ALL RIGHTS RESERVED.
0013 1 *
0014 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0015 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0016 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0017 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0018 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0019 1 * TRANSFERRED.
0020 1 *
0021 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0022 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0023 1 * CORPORATION.
0024 1 *
0025 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0026 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0027 1 *
0028 1 *
0029 1 *****
0030 1
0031 1 ++
0032 1 FACILITY:      EXCHANGE - Foreign volume interchange facility
0033 1
0034 1 ABSTRACT:      Miscellaneous debug and trace routines
0035 1
0036 1 ENVIRONMENT:    VAX/VMS User mode
0037 1
0038 1 AUTHOR:        CW Hobbs                      CREATION DATE: 16-July-1982
0039 1
0040 1 MODIFIED BY:
0041 1
0042 1 V03-002 CWH3002      CW Hobbs                      12-Apr-1984
0043 1 Add routines to dump FAB and NAM.
0044 1
0045 1
0046 1 --
```



```

: 48      0047 1 ! Include files:
: 49      0048 1
: 50      0049 1 MACRO $module_name_string = 'exch$dbg' %;      ! The require file needs to know our module name
: 51      0050 1 REQUIRE 'SRC$:EXCREQ'
: 52      0051 1 ;
: 53      0148 1
: 54      0149 1 PSECT
: 55      0150 1     OWN = exch$dbg_own (WRITE)
: 56      0151 1 ;
: 57      0152 1
: 58      0153 1 %SBTTL 'Module table of contents'
: 59      0154 1
: 60      0155 1 ! Module table of contents:
: 61      0156 1
: 62      0157 1 FORWARD ROUTINE
: 63      0158 1     exch$dbg_dump_dosv : NOVALUE,      ! Format the contents of a DOS11$ block
: 64      0159 1     exch$dbg_fab_dev_print,      ! Print device characteristics from FAB info
: 65      0160 1     exch$dbg_fab_dump,      ! Hex dump of a fab
: 66      0161 1     exch$dbg_fab_name_print,      ! Print the name information attached to a FAB
: 67      0162 1     exch$dbg_nam_dump,      ! Hex dump of a nam
: 68      0163 1     exch$dbg_utl_print_devchar      ! Explode a DEV device characteristics field (e.g. FAB$$_DEV
: 69      0164 1 ;
: 70      0165 1
: 71      0166 1 ! EXCHANGE facility routines
: 72      0167 1
: 73      0168 1 EXTERNAL ROUTINE
: 74      0169 1     exch$dos11_expand_filename,      ! Expand filename info into a context block
: 75      0170 1     exch$util_dos11ctx_allocate,      ! Grab a context block
: 76      0171 1     exch$util_dos11ctx_release : NOVALUE      ! Return the context block
: 77      0172 1 ;
: 78      0173 1
: 79      0174 1 ! Bound declarations:
: 80      0175 1
: 81      0176 1 ! BIND
: 82      0177 1 ;
```



```
84 0178 1 GLOBAL ROUTINE exch$dbg_dump_dosv (volb : $ref_bblock, flags) : NOVALUE =
85 0179 2 BEGIN
86 0180 2 $dbgtrc_prefix ('dbg_dump_dosv> ');
87 0181 2
88 0182 2 LOCAL
89 0183 2     ctx : $ref_bblock,
90 0184 2     ent : $ref_bblock,
91 0185 2     lcl : $bblock [4],
92 0186 2     first
93 0187 2 ;
94 0188 2
95 0189 2 BIND
96 0190 2     dosv = volb [volb$a_vfmt_specific] : $ref_bblock
97 0191 2 ;
98 0192 2
99 0193 2 BUILTIN
100 0194 2     ACTUALCOUNT
101 0195 2 ;
102 0196 2
103 0197 2 IF ACTUALCOUNT () EQL 2
104 0198 2 THEN
105 0199 2     lcl [0,0,32,0] = .flags
106 0200 2 ELSE
107 0201 2     lcl [0,0,32,0] = -1;
108 0202 2
109 0203 2 IF .lcl [dmpdsv$v_volb]
110 0204 2 THEN
111 0205 2     $print_fao ('    dump_dosv> entry - volb !XL, dosv !XL', .volb, .dosv);
112 0206 2
113 0207 2 $block_check (2, .volb, volb, 623);
114 0208 2 $block_check (2, .dosv, dos11, 624);
115 0209 2
116 0210 2 IF .lcl [dmpdsv$v_status]
117 0211 2 THEN
118 P 0212 2     $print_fao ('    dump_dosv> status - posvalid !UL, bot !UL, tapemark !UL, eot !UL, dir !UL, curmrkd !UL'
119 P 0213 2         .dosv [dos11$v_position_valid], .dosv [dos11$v_beg_of_tape], .dosv [dos11$v_tape_mark],
120 0214 2         .dosv [dos11$v_end_of_tape], .dosv [dos11$v_directory], .dosv [dos11$v_current_marked]);
121 0215 2
122 0216 2 IF .lcl [dmpdsv$v_position]
123 0217 2 THEN
124 P 0218 2     $print_fao ('    dump_dosv> .dos11$a_current_entry !XL, .dos11$l_current_file !UL',
125 0219 2         .dosv [dos11$a_current_entry], .dosv [dos11$l_current_file]);
126 0220 2
127 0221 2 IF .lcl [dmpdsv$v_entries]
128 0222 2 THEN
129 0223 2     BEGIN
130 0224 2
131 0225 2         ! If the pointer points to the queue header, we have reached the end of the list
132 0226 2         !
133 0227 2         ent = .dosv [dos11$a_entry_flink];
134 0228 2         ctx = exch$util_dos11ctx_allocate (.volb, 0);           ! Grab a context block for the loop
135 0229 2         first = true;
136 0230 2         WHILE .ent NEQ dosv [dos11$q_entry_header]
137 0231 2         DO
138 0232 2             BEGIN
139 0233 2                 IF .first
140 0234 2
```



```
141 0235 4 THEN
142 0236 5 BEGIN
143 0237 5 $print_lit (' dump_dosv> Num Entry Flink Blink Name Block Val');
144 P 0238 5 $print_fao (' dump_dosv> !3UL !XL !XL !XL !13AF !5UL !UL',
145 P 0239 5 9999, dosv [dos11$entry_header], dosv [dos11$a_entry_flink], .dosv [dos11$a_entry_bli
146 0240 5 12, UPLIT BYTE ('queue-header'), 999999, 99);
147 0241 4 END;
148 0242 4 first = false;
149 0243 4
150 0244 4 CH$MOVE (dos11ctx$s_entry, ent [dos11ent$l_filename_1], ctx [dos11ctx$t_entry]);
151 0245 4 exch$dos11_expand_filename (.ctx);
152 0246 4
153 P 0247 4 $print_fao (' dump_dosv> !3UL !XL !XL !XL !13AF !5UL !UL',
154 P 0248 4 .ent [dos11ent$w_file_number], .ent, .ent [dos11ent$a_flink], .ent [dos11ent$a_blin
155 P 0249 4 .ctx [dos11ctx$l_exp_fullname_len], ctx [dos11ctx$t_exp_fullname],
156 0250 4 .ent [dos11ent$w_blocks], .ent [dos11ent$w_blocks_valid]);
157 0251 4
158 0252 4 ent = .ent [dos11ent$a_flink]; ! Move to the next
159 0253 3 END;
160 0254 3
161 0255 3 exch$util_dos11ctx_release (.ctx); ! Return the context block
162 0256 2 END;
163 0257 2
164 0258 2 RETURN;
165 0259 1 END;
```

```
.TITLE EXCH$DBG Conditionally assembled debugging routines
.IDENT \V04-000\
.PSECT EXCH$DBG_PLIT,NOWRT,2

20 3E 76 73 6F 64 5F 70 6D 75 64 20 20 20 20 00000 P.AAB: .ASCII \ dump_dosv> entry - volb !XL, dosv !\
58 21 20 62 6C 6F 76 20 2D 20 79 72 74 6E 65 0000F
20 20 20 20 0001E
6C 61 76 73 6F 70 20 2D 20 73 75 74 61 74 73 00028
20 2C 4C 58 0002C P.AAA: .ASCII \XL\<0><0>
6B 72 61 6D 65 70 61 74 20 2C 4C 55 21 20 64 69 00030 P.AAA: .LONG 17694762
20 2C 4C 55 21 20 74 6F 65 20 2C 4C 55 21 20 72 69 64 00034 P.AAD: .ADDRESS P.AAB
6C 61 76 73 6F 70 20 2D 20 73 75 74 61 74 73 00043 P.AAD: .ASCII \ dump_dosv> status - posvalid !UL, bo\
20 2C 4C 55 21 20 74 6F 65 20 2C 4C 55 21 20 72 69 64 00052
6B 72 61 6D 65 70 61 74 20 2C 4C 55 21 20 74 0005C .ASCII \t !UL, tapemark !UL, eot !UL, dir !UL, c\
20 2C 4C 55 21 20 74 6F 65 20 2C 4C 55 21 20 72 69 64 0006B
00 00 4C 55 21 20 64 6B 72 6D 72 75 0007A
010E005A 00090 P.AAC: .ASCII \urmrkd !UL\<0><0>
20 3E 76 73 6F 64 5F 70 6D 75 64 20 20 20 20 00094 P.AAC: .LONG 17694810
6E 65 72 72 75 63 5F 61 24 31 31 73 6F 64 2E 00098 P.AAF: .ADDRESS P.AAD
5F 6C 24 31 31 73 6F 64 2E 20 20 2C 4C 58 21 000A7 P.AAF: .ASCII \ dump_dosv> .dos11$a_current_entry \
21 20 20 65 6C 69 66 5F 74 6E 65 72 72 75 63 000B6
010E0048 000E0 P.AAE: .ASCII \!XL, .dos11$l_current_file !UL\
20 3E 76 73 6F 64 5F 70 6D 75 64 20 20 20 20 000C0 P.AAE: .LONG 17694792
6E 65 72 72 75 63 5F 61 24 31 31 73 6F 64 2E 000CF P.AAH: .ADDRESS P.AAF
20 3E 76 73 6F 64 5F 70 6D 75 64 20 20 20 20 000DE P.AAH: .ASCII \ dump_dosv> Num Entry Flink \
```



20	20	20	20	79	72	74	6E	45	20	20	20	6D	75	4E	000F7
6D	61	4E	20	20	20	20	20	20	6B	6E	69	6C	46	20	00106
42	20	20	20	20	20	20	20	20	6B	6E	69	6C	42	20	00110
					00	6C	61	56	20	20	6B	63	6F	6C	0011F
													010E004F		0012E
													00000000		00138
20	3E	76	73	6F	64	5F	70	6D	75	64	20	20	20	20	0013C
20	4C	58	21	20	20	4C	58	21	20	20	4C	55	33	21	00140
					33	31	21	20	20	20	4C	58	21	20	0014F
21	20	20	20	20	4C	55	35	21	20	20	20	20	46	41	0015E
									00	00	00	4C	55		00168
													010E0039		00177
													00000000		0017C
20	3E	76	73	6F	64	5F	70	6D	75	64	20	20	20	20	00184
20	4C	58	21	20	20	4C	58	21	20	20	4C	55	33	21	00190
					33	31	21	20	20	20	4C	58	21	20	0019F
21	20	20	20	20	4C	55	35	21	20	20	20	20	46	41	001AE
									00	00	00	4C	55		001B8
													010E0039		001C7
													00000000		001CC
															001D0

Name	Block	Val-
.ASCII \ Blink		
.ASCII \<0>		
P.AAG: .LONG 17694799		
.ADDRESS P.AAH		
P.AAJ: .ASCII \ dump_dosv> !3UL !XL !XL !XL !13\		
.ASCII \AF !5UL !UL\<0><0><0>		
P.AAI: .LONG 17694777		
.ADDRESS P.AAJ		
P.AAK: .ASCII \queue-header\		
P.AAM: .ASCII \ dump_dosv> !3UL !XL !XL !XL !13\		
.ASCII \AF !5UL !UL\<0><0><0>		
P.AAL: .LONG 17694777		
.ADDRESS P.AAM		

.EXTRN EXCH\$DOS11\_EXPAND\_FILENAME  
.EXTRN EXCH\$UTIL\_DOS11CTX\_ALLOCATE  
.EXTRN EXCH\$UTIL\_DOS11CTX\_RELEASE  
.EXTRN EXCH\$UTIL\_FAO\_BUFFER  
.EXTRN LIB\$PUT\_OUTPUT, EXCH\$UTIL\_BLOCK\_CHECK  
.PSECT EXCH\$DBG\_CODE, NOWRT, 2

OFFC 00000

.ENTRY EXCH\$DBG_DUMP_DOSV, Save R2,R3,R4,R5,R6,R7,-;	0178
MOVAB LIB\$PUT_OUTPUT, R11	
ADDL3 #84, VO[B, R4	0190
CMPB (AP), #2	0197
BNEQ 1\$	
MOVL FLAGS, LCL	0199
BRB 2\$	
MNEGL #1, LCL	0201
BLBC LCL, 3\$	0203
PUSHL (R4)	0205
PUSHL VOLB	
PUSHAB P.AAA	
CALLS #3, EXCH\$UTIL_FAO_BUFFER	
PUSHL R0	
CALLS #1, LIB\$PUT_OUTPUT	
MOVL #68878579, R2	0207
MOVZWL #623, R1	
MOVL VOLB, R0	
JSB EXCH\$UTIL_BLOCK_CHECK	0208
MOVL (R4), R7	
MOVL #3539197, R2	
MOVZWL #624, R1	
MOVL R7, R0	
JSB EXCH\$UTIL_BLOCK_CHECK	
BBC #1, LCL, 4\$	0210

54	04	5B	00000000G	00	9E	00002
		AC	00000054	8F	C1	00009
		02		6C	91	00012
				06	12	00015
		53	08	AC	D0	00017
				03	11	0001B
		53		01	CE	0001D
		15		53	E9	00020
				64	DD	00023
			04	AC	DD	00025
			0000	CF	9F	00028
		EF		03	FB	0002C
				50	DD	00033
		6B		01	FB	00035
		52	041B00F3	8F	D0	00038
		51	026F	8F	3C	0003F
		50	04	AC	D0	00044
			00000000G	EF	16	00048
		57		64	D0	0004E
		52	003600FD	8F	D0	00051
		51	0270	8F	3C	00058
		50		57	D0	0005D
			00000000G	EF	16	00060
32		53		01	E1	00066



7E	60	50	0C	A7	9E	0006A	MOVAB	12(R7), R0	0214
7E	60	01		05	EF	0006E	EXTZV	#5, #1, (R0), -(SP)	
7E	60	01		04	EF	00073	EXTZV	#4, #1, (R0), -(SP)	
7E	60	01		03	EF	00078	EXTZV	#3, #1, (R0), -(SP)	
7E	60	01		02	EF	0007D	EXTZV	#2, #1, (R0), -(SP)	
7E	60	01		01	EF	00082	EXTZV	#1, #1, (R0), -(SP)	
7E	60	01		00	EF	00087	EXTZV	#0, #1, (R0), -(SP)	
			0000'	CF	9F	0008C	PUSHAB	P.AAC	
	00000000G	EF		07	FB	00090	CALLS	#7, EXCH\$UTIL_FAO_BUFFER	
				50	DD	00097	PUSHL	R0	
	16	6B		01	FB	00099	CALLS	#1, LIB\$PUT_OUTPUT	
		53		02	E1	0009C	BBC	#2, LCL, 5\$	0216
			0E	A7	DD	000A0	PUSHL	14(R7)	0219
			1A	A7	DD	000A3	PUSHL	26(R7)	
			0000'	CF	9F	000A6	PUSHAB	P.AAE	
	00000000G	EF		03	FB	000AA	CALLS	#3, EXCH\$UTIL_FAO_BUFFER	
				50	DD	000B1	PUSHL	R0	
	01	6B		01	FB	000B3	CALLS	#1, LIB\$PUT_OUTPUT	
		53		03	E0	000B6	BBS	#3, LCL, 6\$	0221
				04	000BA	RET			
	56		12	A7	D0	000BB	MOVL	18(R7), ENT	0227
				7E	D4	000BF	CLRL	-(SP)	0228
			04	AC	DD	000C1	PUSHL	VOLB	
	00000000G	EF		02	FB	000C4	CALLS	#2, EXCH\$UTIL_DOS11CTX_ALLOCATE	
		58		50	D0	000CB	MOVL	R0, CTX	
		59		01	D0	000CE	MOVL	#1, FIRST	0229
		5A		A8	9E	000D1	MOVAB	102(R8), R10	0250
		50		A7	9E	000D5	MOVAB	18(R7), R0	0230
		50		56	D1	000D9	CMPL	ENT, R0	
				74	13	000DC	BEQL	9\$	
	33			59	E9	000DE	BLBC	FIRST, 8\$	0234
			0000'	CF	9F	000E1	PUSHAB	P.AAG	0237
	6B			01	FB	000E5	CALLS	#1, LIB\$PUT_OUTPUT	
	7E		63	8F	9A	000E8	MOVZBL	#99, -(SP)	0240
			000F423F	8F	DD	000EC	PUSHL	#999999	
			0000'	CF	9F	000F2	PUSHAB	P.AAK	
				0C	DD	000F6	PUSHL	#12	
	7E		12	A7	7D	000F8	MOVQ	18(R7), -(SP)	
			12	A7	9F	000FC	PUSHAB	18(R7)	
	7E		270F	8F	3C	000FF	MOVZWL	#9999, -(SP)	
			0000'	CF	9F	00104	PUSHAB	P.AAI	
	00000000G	EF		09	FB	00108	CALLS	#9, EXCH\$UTIL_FAO_BUFFER	
				50	DD	0010F	PUSHL	R0	
	6B			01	FB	00111	CALLS	#1, LIB\$PUT_OUTPUT	
				59	D4	00114	CLRL	FIRST	0242
	3C	A8	08	A6	14	28	MOV3	#20, 8(ENT), 60(CTX)	0244
				58	DD	0011C	PUSHL	CTX	0245
	00000000G	EF		01	FB	0011E	CALLS	#1, EXCH\$DOS11_EXPAND_FILENAME	
7E	1A	A6		00	EF	00125	EXTZV	#0, #1, 26(ENT), -(SP)	0250
				7E	A6	3C	MOVZWL	22(ENT), -(SP)	
				5A	DD	0012F	PUSHL	R10	
			50	A8	DD	00131	PUSHL	80(CTX)	
	7E			66	7D	00134	MOVQ	(ENT), -(SP)	
				56	DD	00137	PUSHL	ENT	
	7E		18	A6	3C	00139	MOVZWL	24(ENT), -(SP)	
			0000'	CF	9F	0013D	PUSHAB	P.AAL	
	00000000G	EF		09	FB	00141	CALLS	#9, EXCH\$UTIL_FAO_BUFFER	



EXCH\$DBG  
V04-000

Conditionally assembled debugging routines  
Module table of contents

6 5  
16-Sep-1984 00:45:26  
14-Sep-1984 12:29:02

VAX-11 Bliss-32 V4.0-742  
[EXCHNG.SRC]EXCDBG.B32;1

Page 7  
(3)

6B	50	DD	00148	PUSHL	R0	:
56	01	FB	0014A	CALLS	#1, LIB\$PUT_OUTPUT	:
	66	DO	0014D	MOVL	(ENT), ENT	: 0252
	83	11	00150	BRB	7\$	: 0230
	58	DD	00152	PUSHL	CTX	: 0255
00000000G EF	01	FB	00154	CALLS	#1, EXCH\$UTIL_DOS11CTX_RELEASE	:
	04	0015B	RET			: 0259

; Routine Size: 348 bytes, Routine Base: EXCH\$DBG\_CODE + 0000

EXC  
V04

: R

```
167 0260 1 GLOBAL ROUTINE exch$dbg_fab_dev_print (fabb : $ref_bblock) = %SBTTL 'exch$dbg_fab_dev_print'
168 0261 2 BEGIN
169 0262
170 0263 $dbgtrc_prefix ('fab_dev_print > ');
171 0264
172 0265 OWN
173 0266     devbufsiz,
174 0267     devchar,
175 0268     devclass,
176 0269     devdepend,
177 0270     devnam      : $bblock [64],
178 0271     devnamlen,
179 0272     devtype,
180 0273     maxblock;
181 0274
182 0275 OWN
183 0276     dev_item : VECTOR [22, LONG] INITIAL (
184 0277         WORD (dvi$ devbufsiz, 4),          LONG (devbufsiz, 0),
185 0278         (dvi$ devchar^16 OR 4),          devchar,      0,
186 0279         (dvi$ devclass^16 OR 4),         devclass,      0,
187 0280         (dvi$ devdepend^16 OR 4),        devdepend,    0,
188 0281         (dvi$ fulldevnam^16 OR 64),      devnam,        devnamlen,
189 0282         (dvi$ devtype^16 OR 4),          devtype,      0,
190 0283         (dvi$ maxblock^16 OR 4),         maxblock,     0,
191 0284         0);
192 0285
193 0286 LOCAL
194 0287     status,
195 0288     inp_desc : $desc_block,
196 0289     tmp_desc : $desc_block,
197 0290     nam : $ref_bblock;
198 0291
199 0292     nam = .fabb [fab$l_nam];
200 0293     $stat_str_desc_init (inp_desc, .nam [nam$b_dev], .nam [nam$l_dev]);
201 0294
202 0295     status = $getdviw (efn=0, devnam=inp_desc, itmlst=dev_item);
203 0296     IF NOT .status
204 0297     THEN
205 0298     BEGIN
206 0299         $print_fao ('***** Error in $GETDVI for device "'AS'" *****', inp_desc);
207 0300         status = ((.status AND %X'FFFFFFF8') OR 1);
208 0301         SIGNAL ($warning_stat (status));
209 0302         RETURN 0;
210 0303     END;
211 0304
212 0305     tmp_desc [dsc$w_length] = .devnamlen;
213 0306     tmp_desc [dsc$a_pointer] = devnam;
214 0307     $print_fao ('Getdvi for name "'AS'" resolved to device "'AS'", inp_desc, tmp_desc);
215 0308     $print_fao ('Bufsiz = !UL, Maxblocks = !UL, Class = !XB, Type = !XB', .devbufsiz, .maxblock, .devclass, .
216 0309     BEGIN
217 0310     BIND
218 0311         dep = devdepend : $bblock,
219 0312         dev = devchar   : $bblock;
220 P 0313     $print_fao ('Cylinders = !UL, Tracks = !UL, Sectors = !UL, DevChar = !XL'
221 0314         .dep[0,16,16,0], .dep[0,8,8,0], .dep[0,0,8,0], .dev [0,0,32,0]);
222 0315
223 0316     END;
```



```
: 224      0317 2 exch$dbg_utl_print_devchar (.devchar);  
: 225      0318 2  
: 226      0319 2 RETURN 1;  
: 227      0320 1 END;
```

.PSECT EXCH\$DBG\_OWN,NOEXE,2

00000 DEVBUFSIZ:

.BLKB 4

00004 DEVCHAR: .BLKB 4

00008 DEVCLASS:

.BLKB 4

0000C DEVDEPEND:

.BLKB 4

00010 DEVNAM: .BLKB 64

00050 DEVNAMLEN:

.BLKB 4

00054 DEVTYPE: .BLKB 4

00058 MAXBLOCK:

.BLKB 4

0004 0008 0005C DEV\_ITEM:

.WORD 8, 4

00000000' 00060

.ADDRESS DEVBUFSIZ

00000000' 00064

.LONG 0

00020004' 00068

.LONG 131076

00000000' 0006C

.ADDRESS DEVCHAR

00040004' 00070

.LONG 0, 262148

00000000' 00078

.ADDRESS DEVCLASS

000A0004' 0007C

.LONG 0, 655364

00000000' 00084

.ADDRESS DEVDEPEND

00E80040' 00088

.LONG 0, 15204416

00000000' 00090

.ADDRESS DEVNAM, DEVNAMLEN

00060004' 00098

.LONG 393220

00000000' 0009C

.ADDRESS DEVTYPE

001A0004' 000A0

.LONG 0, 1703940

00000000' 000A8

.ADDRESS MAXBLOCK

00000000' 000AC

.LONG 0, 0

.PSECT EXCH\$DBG\_PLIT,NOWRT,2

```
6E 69 20 72 6F 72 72 45 20 20 2A 2A 2A 2A 2A 001D4 P.AAO: .ASCII \***** Error in $GETDVI for device '!AS'\  
65 64 20 72 6F 66 20 49 56 44 54 45 47 24 20 001E3  
22 53 41 21 22 20 65 63 69 76 001F2  
00 2A 2A 2A 2A 2A 20 20 001FC  
010E002F' 00204 P.AAN: .ASCII \ *****\<0>  
00000000' 00208 .LONG 17694767  
65 6D 61 6E 20 72 6F 66 20 69 76 64 74 65 47 0020C P.AAQ: .ADDRESS P.AAO  
64 65 76 6C 6F 73 65 72 20 22 53 41 21 22 20 0021B .ASCII \Getdvi for name '!AS' resolved to device\  
65 63 69 76 65 64 20 6F 74 20 0022A  
00 00 22 53 41 21 22 20 00234  
010E002E' 0023C P.AAP: .ASCII \ '!AS'\<0><0>  
00000000' 00240 .LONG 17694766  
20 20 2C 4C 55 21 20 3D 20 7A 69 73 66 75 42 00244 P.AAS: .ADDRESS P.AAQ  
4C 55 21 20 3D 20 73 6B 63 6F 6C 62 78 61 4D 00253 .ASCII \Bufsiz = !UL, Maxblocks = !UL, Class =\  
3D 20 73 73 61 6C 43 20 20 2C 00262
```



EXCH\$DBG  
V04-000

Conditionally assembled debugging routines  
exch\$dbg\_fab\_dev\_print

J 5  
16-Sep-1984 00:45:26  
14-Sep-1984 12:29:02

VAX-11 Bliss-32 V4.0-742  
[EXCHNG.SRC]EXCDBG.B32;1

Page 10  
(4)

```
21 20 3D 20 65 70 79 54 20 20 2C 42 58 21 20 0026C
00 00 00 42 58 0027B
010E0039 00280 P.AAR: .LONG 17694777
00000000' 00284 .ADDRESS P.AAS
4C 55 21 20 3D 20 73 72 65 64 6E 69 6C 79 43 00288 P.AAU: .ASCII \Cylinders = !UL, Tracks = !UL, Sectors =\
2C 4C 55 21 20 3D 20 73 68 63 61 72 54 20 2C 00297
3D 20 72 61 68 43 76 65 44 20 2C 4C 55 21 20 002A6
00 4C 58 21 20 002B0
010E003B 002C4 P.AAT: .LONG 17694779
00000000' 002C8 .ADDRESS P.AAU
```

DEP=  
DEV=

DEVDEPEND  
DEVCHAR

.EXTRN SYSS\$GETDVIW

.PSECT EXCH\$DBG\_CODE,NOWRT,2

.ENTRY EXCH\$DBG FAB DEV PRINT, Save R2,R3,R4,R5

MOVAB LIB\$PUT\_OUTPUT, R5

MOVAB EXCH\$UTIL\_FAO\_BUFFER, R4

MOVAB DEP, R3

SUBL2 #16, SP

MOVL FABB, R0

MOVL 40(R0), NAM

MOVW #270, DESC+2

MOVZBW 57(NAM), DESC

MOVL 68(NAM), DESC+4

CLRQ -(SP)

CLRQ -(SP)

PUSHAB DEV\_ITEM

PUSHAB INP\_DESC

CLRQ -(SP)

CALLS #8, SYSS\$GETDVIW

MOVL R0, STATUS

BLBS STATUS, 1\$

PUSHAB INP\_DESC

PUSHAB P.AAN

CALLS #2, EXCH\$UTIL\_FAO\_BUFFER

PUSHL R0

CALLS #1, LIB\$PUT\_OUTPUT

BICL3 #7, STATUS, R0

BISL3 #1, R0, STATUS

BICB2 #7, STATUS2

PUSHL STATUS2

CALLS #1, LIB\$SIGNAL

BRB 2\$

MOVW DEVNAMLEN, TMP\_DESC

MOVAB DEVNAM, TMP\_DESC+4

PUSHL SP

PUSHAB INP\_DESC

PUSHAB P.AAP

CALLS #3, EXCH\$UTIL\_FAO\_BUFFER

PUSHL R0

CALLS #1, LIB\$PUT\_OUTPUT

PUSHL DEVTYPE

PUSHL DEVCLASS

0260

0292

0293

0295

0296

0299

0300

0301

0302

0305

0306

0307

0308

EXC  
V04

: R



EXCH\$DBG  
V04-000

Conditionally assembled debugging routines  
exch\$dbg\_fab\_dev\_print

K 5  
16-Sep-1984 00:45:26  
14-Sep-1984 12:29:02

VAX-11 Bliss-32 V4.0-742  
[EXCHNG.SRC]EXCDBG.B32;1

Page 11  
(4)

	4C	A3	DD	0008E	PUSHL	MAXBLOCK	:
	F4	A3	DD	00091	PUSHL	DEVBUFSIZ	:
	0000'	CF	9F	00094	PUSHAB	P.AAR	:
64		05	FB	00098	CALLS	#5, EXCH\$UTIL_FAO_BUFFER	:
		50	DD	0009B	PUSHL	R0	:
65		01	FB	0009D	CALLS	#1, LIB\$PUT_OUTPUT	:
	F8	A3	DD	000A0	PUSHL	DEV	: 0314
7E		63	9A	000A3	MOVZBL	DEP, -(SP)	:
7E	01	A3	9A	000A6	MOVZBL	DEP+1, -(SP)	:
7E	02	A3	3C	000AA	MOVZWL	DEP+2, -(SP)	:
	0000'	CF	9F	000AE	PUSHAB	P.AAT	:
64		05	FB	000B2	CALLS	#5, EXCH\$UTIL_FAO_BUFFER	:
		50	DD	000B5	PUSHL	R0	:
65		01	FB	000B7	CALLS	#1, LIB\$PUT_OUTPUT	:
	F8	A3	DD	000BA	PUSHL	DEVCHAR	: 0317
0000V	CF	01	FB	000BD	CALLS	#1, EXCH\$DBG_UTL_PRINT_DEVCHAR	:
	50	01	D0	000C2	MOVL	#1, R0	: 0319
			04	000C5	RET		:
		50	D4	000C6 2\$:	CLRL	R0	: 0320
			04	000C8	RET		:

; Routine Size: 201 bytes, Routine Base: EXCH\$DBG\_CODE + 015C

```
229 0321 1 GLOBAL ROUTINE exch$dbg_fab_dump (fab : $ref_bblock) = %SBTTL 'exch$dbg_fab_dump'
230 0322 2 BEGIN
231 0323 2
232 0324 2 $dbgtrc_prefix ('fab_dump> ');
233 0325 2
234 0326 2 $sprintf_fao (' Dump of fab at !XL:', .fab);
235 0327 2 ! $sprintf_fao (' !15AS !XB - block id ', %ASCID 'FAB$B BID', .fab[FAB$B BID]);
236 0328 2 ! $sprintf_fao (' !15AS !XB - block len ', %ASCID 'FAB$B BLN', .fab[FAB$B BLN]);
237 0329 2 ! $sprintf_fao (' !15AS !XW - internal file index ', %ASCID 'FAB$W IFI', .fab[FAB$W IFI]);
238 0330 2 $sprintf_fao (' !15AS !XL - file options ', %ASCID 'FAB$L FOP', .fab[FAB$L FOP]);
239 0331 2 $sprintf_fao (' !15AS !XL - status ', %ASCID 'FAB$L STS', .fab[FAB$L STS]);
240 0332 2 $sprintf_fao (' !15AS !XL - status value ', %ASCID 'FAB$L STV', .fab[FAB$L STV]);
241 0333 2 ! $sprintf_fao (' !15AS !XL - allocation quantity ', %ASCID 'FAB$L ALQ', .fab[FAB$L ALQ]);
242 0334 2 ! $sprintf_fao (' !15AS !XW - default allocation quantity ', %ASCID 'FAB$W DEQ', .fab[FAB$W DEQ]);
243 0335 2 $sprintf_fao (' !15AS !XB - file access ', %ASCID 'FAB$B FAC', .fab[FAB$B FAC]);
244 0336 2 $sprintf_fao (' !15AS !XB - file sharing ', %ASCID 'FAB$B SHR', .fab[FAB$B SHR]);
245 0337 2 $sprintf_fao (' !15AS !XL - user context ', %ASCID 'FAB$L CTX', .fab[FAB$L CTX]);
246 0338 2 $sprintf_fao (' !15AS !XB - file organization ', %ASCID 'FAB$B ORG', .fab[FAB$B ORG]);
247 0339 2 $sprintf_fao (' !15AS !XB - record format ', %ASCID 'FAB$B RAT', .fab[FAB$B RAT]);
248 0340 2 $sprintf_fao (' !15AS !XB - record format ', %ASCID 'FAB$B RFM', .fab[FAB$B RFM]);
249 P 0341 2 $sprintf_fao (' !15AS !XB !XL '!AF' ', %ASCID 'filename',
250 0342 2 .fab[FAB$B FNS], .fab[FAB$L FNA], .fab[fab$b_fns], .fab[fab$l_fna]);
251 P 0343 2 $sprintf_fao (' !15AS !XB !XL '!AF' ', %ASCID 'default name',
252 0344 2 .fab[FAB$B DNS], .fab[FAB$L DNA], .fab[fab$b_dns], .fab[fab$l_dna]);
253 0345 2 $sprintf_fao (' !15AS !XL - device characteristics ', %ASCID 'FAB$[DEV]', .fab[FAB$[DEV]]);
254 0346 2 exch$dbg_util_print_devchar (.fab[FAB$L DEV]);
255 0347 2 $sprintf_fao (' !15AS !XL - nam block address ', %ASCID 'FAB$L NAM', .fab[FAB$L NAM]);
256 0348 2 IF .fab[fab$l_nam] NEQ 0
257 0349 2 THEN exch$dbg_nam_dump (.fab[fab$l_nam]);
258 0350 2
259 0351 2 RETURN 1;
260 0352 1 END;
```

```
.PSECT EXCH$DBG_PLIT,NOWRT,2
62 61 66 20 66 6F 20 70 6D 75 44 20 20 20 20 002CC P.AAW: .ASCII \ Dump of fab at !XL:\<0>
20 20 20 20 20 20 20 20 20 20 20 20 20 20 002DB
20 20 20 20 20 20 20 20 20 20 20 20 20 20 002E4 P.AAV: .LONG 17694743
00 00 20 20 78 65 64 6E 69 20 65 6C 69 66 20 002E8 .ADDRESS P.AAW
20 20 20 20 20 20 20 20 20 20 20 20 20 20 002EC P.AAY: .ASCII \ !15AS !XW - internal\
20 20 20 20 20 20 20 20 20 20 20 20 20 20 002FB
00 00 20 20 78 65 64 6E 69 20 65 6C 69 66 20 0030A
20 20 20 20 20 20 20 20 20 20 20 20 20 20 00314 .ASCII \ file index \<0><0><0>
20 20 20 20 20 20 20 20 20 20 20 20 20 20 00323
20 20 20 20 20 20 20 20 20 20 20 20 20 20 00324 P.AAX: .LONG 17694773
20 20 20 20 20 20 20 20 20 20 20 20 20 20 00328 .ADDRESS P.AAY
20 20 20 20 20 20 20 20 20 20 20 20 20 20 0032C P.ABA: .ASCII \FAB$W IFI\<0><0><0>
20 20 20 20 20 20 20 20 20 20 20 20 20 20 00338 P.AAZ: .LONG 17694729
20 20 20 20 20 20 20 20 20 20 20 20 20 20 0033C .ADDRESS P.ABA
20 20 20 20 20 20 20 20 20 20 20 20 20 20 00340 P.ABC: .ASCII \ !15AS !XL - file options\
20 20 20 20 20 20 20 20 20 20 20 20 20 20 0034F
20 20 20 20 20 20 20 20 20 20 20 20 20 20 0035E
20 20 20 20 20 20 20 20 20 20 20 20 20 20 00368 .ASCII \ \<0><0>
20 20 20 20 20 20 20 20 20 20 20 20 20 20 0036C P.ABB: .LONG 17694762
20 20 20 20 20 20 20 20 20 20 20 20 20 20 00370 .ADDRESS P.ABC
```



EXCH\$DBG  
V04-000

Conditionally assembled debugging routines  
exch\$dbg\_fab\_dump

M 5  
16-Sep-1984 00:45:26  
14-Sep-1984 12:29:02

VAX-11 Bliss-32 V4.0-742  
[EXCHNG.SRC]EXCDBG.B32;1

Page 13  
(5)

00	00	00	50	4F	46	5F	4C	24	42	41	46	00374	P.ABE:	.ASCII	\FAB\$L_FOP\<0><0><0>	
										010E0009	00380	P.ABD:	.LONG	17694729		
20	20	53	41	35	31	21	20	20	20	20	20	00000000	00384	.ADDRESS	P.ABE	
74	73	20	2D	20	20	20	20	4C	58	21	20	00388	P.ABG:	.ASCII	\	!15AS !XL - status \
									20	20	73	00397				
									20	20	75	003A6				
									010E0024	003AC	P.ABF:	.LONG	17694756			
									00000000	003B0	.ADDRESS	P.ABG				
00	00	00	53	54	53	5F	4C	24	42	41	46	003B4	P.ABI:	.ASCII	\FAB\$L_STS\<0><0><0>	
									010E0009	003C0	P.ABH:	.LONG	17694729			
20	20	53	41	35	31	21	20	20	20	20	20	003C4	.ADDRESS	P.ABI		
74	73	20	2D	20	20	20	20	4C	58	21	20	003C8	P.ABK:	.ASCII	\	!15AS !XL - status value\
					65	75	6C	61	76	20	73	003D7				
									00	00	00	003E6				
									010E002A	003F0		.ASCII	\	\<0><0>		
									00000000	003F4	P.ABJ:	.LONG	17694762			
00	00	00	56	54	53	5F	4C	24	42	41	46	003F8	.ADDRESS	P.ABK		
									010E0009	00408	P.ABM:	.ASCII	\FAB\$L_STV\<0><0><0>			
									00000000	0040C	P.ABL:	.LONG	17694729			
20	20	53	41	35	31	21	20	20	20	20	20	00410	.ADDRESS	P.ABM		
20	20	20	20	20	20	20	20	42	58	21	20	0041F	P.ABO:	.ASCII	\	!15AS !XB - file a\
					61	20	65	6C	69	66	20	0042E				
							00	20	20	73	73	00438		.ASCII	\ccess \<0>	
									010E002F	00440	P.ABN:	.LONG	17694767			
									00000000	00444	.ADDRESS	P.ABO				
00	00	00	43	41	46	5F	42	24	42	41	46	00448	P.ABQ:	.ASCII	\FAB\$L_FAC\<0><0><0>	
									010E0009	00454	P.ABP:	.LONG	17694729			
20	20	53	41	35	31	21	20	20	20	20	20	00458	.ADDRESS	P.ABQ		
20	20	20	20	20	20	20	20	42	58	21	20	0045C	P.ABS:	.ASCII	\	!15AS !XB - file s\
					73	20	65	6C	69	66	20	0046B				
							20	20	67	6E	69	0047A				
									010E0030	00484		.ASCII	\haring \			
									00000000	0048C	P.ABR:	.LONG	17694768			
00	00	00	52	48	53	5F	42	24	42	41	46	00490	.ADDRESS	P.ABS		
									010E0009	00494	P.ABU:	.ASCII	\FAB\$L_SHR\<0><0><0>			
20	20	53	41	35	31	21	20	20	20	20	20	004A0	P.ABT:	.LONG	17694729	
73	75	20	2D	20	20	20	20	4C	58	21	20	004A4	.ADDRESS	P.ABU		
					74	78	65	74	6E	6F	63	004A8	P.ABW:	.ASCII	\	!15AS !XL - user context\
									00	00	00	004B7				
									010E002A	004D0		.ASCII	\	\<0><0>		
									00000000	004D4	P.ABV:	.LONG	17694762			
00	00	00	58	54	43	5F	4C	24	42	41	46	004D8	.ADDRESS	P.ABW		
									010E0009	004DC	P.ABY:	.ASCII	\FAB\$L_CTX\<0><0><0>			
20	20	53	41	35	31	21	20	20	20	20	20	004E8	P.ABX:	.LONG	17694729	
20	20	20	20	20	20	20	20	42	58	21	20	004EC	.ADDRESS	P.ABY		
					6F	20	65	6C	69	66	20	004F0	P.ACA:	.ASCII	\	!15AS !XB - file o\
00	00	20	20	6E	6F	69	74	61	7A	69	6E	004FF				
									010E0009	0050E		.ASCII	\organization \<0><0><0>			
									00000000	00518						
									010E0035	00527						
									00000000	00528	P.ABZ:	.LONG	17694773			
00	00	00	47	52	4F	5F	42	24	42	41	46	0052C	.ADDRESS	P.ACA		
									010E0009	00530	P.ACC:	.ASCII	\FAB\$L_ORG\<0><0><0>			
									00000000	0053C	P.ACB:	.LONG	17694729			
												00540	.ADDRESS	P.ACC		

EXCH\$DBG  
V04-000

Conditionally assembled debugging routines  
exch\$dbg\_fab\_dump

N 5  
16-Sep-1984 00:45:26  
14-Sep-1984 12:29:02

VAX-11 Bliss-32 V4.0-742  
[EXCHNG.SRC]EXCDBG.B32;1

Page 14  
(5)

```
20 20 53 41 35 31 21 20 20 20 20 20 20 20 00544 P.ACE: .ASCII \ !15AS !XB - record\
20 20 20 20 20 20 20 20 20 20 20 20 20 20 00553
00 00 00 20 20 74 61 6D 72 6F 66 20 00562
010E0031 00578 P.ACD: .ASCII \ format \<0><0><0>
00000000' 0057C .LONG 17694769
00 00 00 54 41 52 5F 42 24 42 41 46 00580 P.ACG: .ASCII \FAB$B_RAT\<0><0><0>
010E0009 0058C P.ACF: .LONG 17694729
00000000' 00590 P.ACI: .ADDRESS P.ACG
20 20 53 41 35 31 21 20 20 20 20 20 20 20 00594 P.ACI: .ASCII \ !15AS !XB - record\
20 20 20 20 20 20 20 20 20 20 20 20 20 20 005A3
00 00 00 20 20 74 61 6D 72 6F 66 20 005B2
010E0031 005BC P.ACH: .ASCII \ format \<0><0><0>
00000000' 005CC .LONG 17694769
00 00 00 4D 46 52 5F 42 24 42 41 46 005D0 P.ACK: .ADDRESS P.ACI
010E0009 005DC P.ACJ: .ASCII \FAB$B_RFM\<0><0><0>
00000000' 005E0 P.ACM: .LONG 17694729
20 20 53 41 35 31 21 20 20 20 20 20 20 20 005E4 P.ACM: .ADDRESS P.ACK
41 21 22 20 20 20 20 4C 58 21 20 42 58 005F3 P.ACM: .ASCII \ !15AS !XB !XL '!AF'\
22 46 00602
010E0020 00604 P.ACL: .LONG 17694752
00000000' 00608 .ADDRESS P.ACM
65 6D 61 6E 65 6C 69 66 0060C P.ACO: .ASCII \filename\
010E0008 00614 P.ACN: .LONG 17694728
00000000' 00618 P.ACQ: .ADDRESS P.ACO
20 20 53 41 35 31 21 20 20 20 20 20 20 20 0061C P.ACQ: .ASCII \ !15AS !XB !XL '!AF'\
41 21 22 20 20 20 20 4C 58 21 20 42 58 0062B
22 46 0063A
010E0020 0063C P.ACP: .LONG 17694752
00000000' 00640 .ADDRESS P.ACQ
65 6D 61 6E 20 74 6C 75 61 66 65 64 00644 P.ACS: .ASCII \default name\
010E000C 00650 P.ACR: .LONG 17694732
00000000' 00654 P.ACU: .ADDRESS P.ACS
20 20 53 41 35 31 21 20 20 20 20 20 20 20 00658 P.ACU: .ASCII \ !15AS !XL - device chara\
65 64 20 20 20 20 20 20 4C 58 21 20 20 20 00667
20 20 73 63 69 74 73 69 72 65 74 63 00676
010E0034 00680 P.ACT: .ASCII \cteristics \
00000000' 0068C .LONG 17694772
00 00 00 56 45 44 5F 4C 24 42 41 46 00690 P.ACW: .ADDRESS P.ACU
010E0009 00694 P.ACV: .ASCII \FAB$L_DEV\<0><0><0>
00000000' 006A0 P.ACV: .LONG 17694729
006A4 P.ACV: .ADDRESS P.ACW
20 20 53 41 35 31 21 20 20 20 20 20 20 20 006A8 P.ACY: .ASCII \ !15AS !XL - nam block ad\
61 6E 20 20 20 20 20 20 4C 58 21 20 20 20 006B7
64 61 20 6B 63 6F 6C 62 20 6D 006C6
00 20 20 73 73 65 72 64 006D0
010E002F 006D8 P.ACX: .ASCII \dress \<0>
00000000' 006DC .LONG 17694767
00 00 00 4D 41 4E 5F 4C 24 42 41 46 006E0 P.ADA: .ADDRESS P.ACY
010E0009 006EC P.ACZ: .ASCII \FAB$L_NAM\<0><0><0>
00000000' 006F0 .LONG 17694729
006F0 P.ACZ: .ADDRESS P.ADA
```

.PSECT EXCH\$DBG\_CODE,NOWRT,2

EXC  
V04

20  
61



		001C 00000	.ENTRY	EXCH\$DBG FAB_DUMP, Save R2,R3,R4	: 0321
54	00000000G	00 9E 00002	MOVAB	LIB\$PUT_OUTPUT, R4	
53	00000000G	EF 9E 00009	MOVAB	EXCH\$UTIL_FAO_BUFFER, R3	
52	04	AC D0 00010	MOVL	FAB, R2	: 0326
		52 DD 00014	PUSHL	R2	
	0000'	CF 9F 00016	PUSHAB	P.AAV	
63		02 FB 0001A	CALLS	#2, EXCH\$UTIL_FAO_BUFFER	
		50 DD 0001D	PUSHL	R0	
64		01 FB 0001F	CALLS	#1, LIB\$PUT_OUTPUT	
7E	02	A2 3C 00022	MOVZWL	2(R2), -(SP)	: 0329
	0000'	CF 9F 00026	PUSHAB	P.AAZ	
	0000'	CF 9F 0002A	PUSHAB	P.AAX	
63		03 FB 0002E	CALLS	#3, EXCH\$UTIL_FAO_BUFFER	
		50 DD 00031	PUSHL	R0	
64		01 FB 00033	CALLS	#1, LIB\$PUT_OUTPUT	
	04	A2 DD 00036	PUSHL	4(R2)	: 0330
	0000'	CF 9F 00039	PUSHAB	P.ABD	
	0000'	CF 9F 0003D	PUSHAB	P.ABB	
63		03 FB 00041	CALLS	#3, EXCH\$UTIL_FAO_BUFFER	
		50 DD 00044	PUSHL	R0	
64		01 FB 00046	CALLS	#1, LIB\$PUT_OUTPUT	
	08	A2 DD 00049	PUSHL	8(R2)	: 0331
	0000'	CF 9F 0004C	PUSHAB	P.ABH	
	0000'	CF 9F 00050	PUSHAB	P.ABF	
63		03 FB 00054	CALLS	#3, EXCH\$UTIL_FAO_BUFFER	
		50 DD 00057	PUSHL	R0	
64		01 FB 00059	CALLS	#1, LIB\$PUT_OUTPUT	
	0C	A2 DD 0005C	PUSHL	12(R2)	: 0332
	0000'	CF 9F 0005F	PUSHAB	P.ABL	
	0000'	CF 9F 00063	PUSHAB	P.ABJ	
63		03 FB 00067	CALLS	#3, EXCH\$UTIL_FAO_BUFFER	
		50 DD 0006A	PUSHL	R0	
64		01 FB 0006C	CALLS	#1, LIB\$PUT_OUTPUT	
7E	16	A2 9A 0006F	MOVZBL	22(R2), -(SP)	: 0335
	0000'	CF 9F 00073	PUSHAB	P.ABP	
	0000'	CF 9F 00077	PUSHAB	P.ABN	
63		03 FB 0007B	CALLS	#3, EXCH\$UTIL_FAO_BUFFER	
		50 DD 0007E	PUSHL	R0	
64		01 FB 00080	CALLS	#1, LIB\$PUT_OUTPUT	
7E	17	A2 9A 00083	MOVZBL	23(R2), -(SP)	: 0336
	0000'	CF 9F 00087	PUSHAB	P.ABT	
	0000'	CF 9F 0008B	PUSHAB	P.ABR	
63		03 FB 0008F	CALLS	#3, EXCH\$UTIL_FAO_BUFFER	
		50 DD 00092	PUSHL	R0	
64		01 FB 00094	CALLS	#1, LIB\$PUT_OUTPUT	
	18	A2 DD 00097	PUSHL	24(R2)	: 0337
	0000'	CF 9F 0009A	PUSHAB	P.ABX	
	0000'	CF 9F 0009E	PUSHAB	P.ABV	
63		03 FB 000A2	CALLS	#3, EXCH\$UTIL_FAO_BUFFER	
		50 DD 000A5	PUSHL	R0	
64		01 FB 000A7	CALLS	#1, LIB\$PUT_OUTPUT	
7E	1D	A2 9A 000AA	MOVZBL	29(R2), -(SP)	: 0338
	0000'	CF 9F 000AE	PUSHAB	P.ACB	
	0000'	CF 9F 000B2	PUSHAB	P.ABZ	
63		03 FB 000B6	CALLS	#3, EXCH\$UTIL_FAO_BUFFER	
		50 DD 000B9	PUSHL	R0	
64		01 FB 000BB	CALLS	#1, LIB\$PUT_OUTPUT	



7E	1E	A2	9A	000BE	MOVZBL	30(R2), -(SP)	0339
	0000'	CF	9F	000C2	PUSHAB	P.ACF	
	0000'	CF	9F	000C6	PUSHAB	P.ACD	
63		03	FB	000CA	CALLS	#3, EXCH\$UTIL_FAO_BUFFER	
		50	DD	000CD	PUSHL	R0	
64		01	FB	000CF	CALLS	#1, LIB\$PUT_OUTPUT	
7E	1F	A2	9A	000D2	MOVZBL	31(R2), -(SP)	0340
	0000'	CF	9F	000D6	PUSHAB	P.ACJ	
	0000'	CF	9F	000DA	PUSHAB	P.ACH	
63		03	FB	000DE	CALLS	#3, EXCH\$UTIL_FAO_BUFFER	
		50	DD	000E1	PUSHL	R0	
64		01	FB	000E3	CALLS	#1, LIB\$PUT_OUTPUT	
	2C	A2	DD	000E6	PUSHL	44(R2)	0342
7E	34	A2	9A	000E9	MOVZBL	52(R2), -(SP)	
	2C	A2	DD	000ED	PUSHL	44(R2)	
7E	34	A2	9A	000F0	MOVZBL	52(R2), -(SP)	
	0000'	CF	9F	000F4	PUSHAB	P.ACN	
	0000'	CF	9F	000F8	PUSHAB	P.ACL	
63		06	FB	000FC	CALLS	#6, EXCH\$UTIL_FAO_BUFFER	
		50	DD	000FF	PUSHL	R0	
64		01	FB	00101	CALLS	#1, LIB\$PUT_OUTPUT	
	30	A2	DD	00104	PUSHL	48(R2)	0344
7E	35	A2	9A	00107	MOVZBL	53(R2), -(SP)	
	30	A2	DD	0010B	PUSHL	48(R2)	
7E	35	A2	9A	0010E	MOVZBL	53(R2), -(SP)	
	0000'	CF	9F	00112	PUSHAB	P.ACR	
	0000'	CF	9F	00116	PUSHAB	P.ACP	
63		06	FB	0011A	CALLS	#6, EXCH\$UTIL_FAO_BUFFER	
		50	DD	0011D	PUSHL	R0	
64		01	FB	0011F	CALLS	#1, LIB\$PUT_OUTPUT	
	40	A2	DD	00122	PUSHL	64(R2)	0345
	0000'	CF	9F	00125	PUSHAB	P.ACV	
	0000'	CF	9F	00129	PUSHAB	P.ACT	
63		03	FB	0012D	CALLS	#3, EXCH\$UTIL_FAO_BUFFER	
		50	DD	00130	PUSHL	R0	
64		01	FB	00132	CALLS	#1, LIB\$PUT_OUTPUT	
	40	A2	DD	00135	PUSHL	64(R2)	0346
0000V	CF	01	FB	00138	CALLS	#1, EXCH\$DBG_UTL_PRINT_DEVCHAR	
	28	A2	DD	0013D	PUSHL	40(R2)	0347
	0000'	CF	9F	00140	PUSHAB	P.ACZ	
	0000'	CF	9F	00144	PUSHAB	P.ACX	
63		03	FB	00148	CALLS	#3, EXCH\$UTIL_FAO_BUFFER	
		50	DD	0014B	PUSHL	R0	
64		01	FB	0014D	CALLS	#1, LIB\$PUT_OUTPUT	
	28	A2	D5	00150	TSTL	40(R2)	0348
		08	13	00153	BEQL	1\$	
	28	A2	DD	00155	PUSHL	40(R2)	0349
0000V	CF	01	FB	00158	CALLS	#1, EXCH\$DBG_NAM_DUMP	
	50	01	D0	0015D	MOVL	#1, R0	0351
		04	00160	1\$: RET			0352

; Routine Size: 353 bytes, Routine Base: EXCH\$DBG\_CODE + 0225



```
: 262 0353 1 GLOBAL ROUTINE exch$dbg_fab_name_print (fabb : $ref_bblock) = %SBTTL 'exch$dbg_fab_name_print'
: 263 0354 2 BEGIN
: 264 0355 2
: 265 0356 2 $dbgtrc_prefix ('fab_name_print> ');
: 266 0357 2
: 267 0358 2 LOCAL
: 268 0359 2     tmp_desc : $desc_block,      ! A descriptor for the total file name
: 269 0360 2     dev_desc : $desc_block,    ! A descriptor for the device name
: 270 0361 2     dir_desc : $desc_block,   ! A descriptor for the directory name
: 271 0362 2     dvi_desc : $desc_block,   ! A descriptor for the full device name
: 272 0363 2     nam_desc : $desc_block,    ! A descriptor for the name name
: 273 0364 2     nod_desc : $desc_block,   ! A descriptor for the node name
: 274 0365 2     typ_desc : $desc_block,   ! A descriptor for the type name
: 275 0366 2     ver_desc : $desc_block;    ! A descriptor for the version number
: 276 0367 2
: 277 0368 2 BIND
: 278 0369 2     nam = .fabb [fab$l_nam] : $bblock; ! Get pointer to the name block
: 279 0370 2
: 280 0371 2     tmp_desc [0,0,32,0] = dsc$k_class_s^24 OR dsc$k_dtype_t^16; ! Static desc
: 281 0372 2     dev_desc [0,0,32,0] = dsc$k_class_s^24 OR dsc$k_dtype_t^16; ! Static desc
: 282 0373 2     dir_desc [0,0,32,0] = dsc$k_class_s^24 OR dsc$k_dtype_t^16; ! Static desc
: 283 0374 2     dvi_desc [0,0,32,0] = dsc$k_class_s^24 OR dsc$k_dtype_t^16; ! Static desc
: 284 0375 2     nam_desc [0,0,32,0] = dsc$k_class_s^24 OR dsc$k_dtype_t^16; ! Static desc
: 285 0376 2     nod_desc [0,0,32,0] = dsc$k_class_s^24 OR dsc$k_dtype_t^16; ! Static desc
: 286 0377 2     typ_desc [0,0,32,0] = dsc$k_class_s^24 OR dsc$k_dtype_t^16; ! Static desc
: 287 0378 2     ver_desc [0,0,32,0] = dsc$k_class_s^24 OR dsc$k_dtype_t^16; ! Static desc
: 288 0379 2
: 289 0380 2 IF .fabb [fab$b_fns] NEQ 0
: 290 0381 2 THEN
: 291 0382 3     BEGIN
: 292 0383 3         tmp_desc [dsc$w_length] = .fabb [fab$b_fns]; ! Create file name desc
: 293 0384 3         tmp_desc [dsc$a_pointer] = .fabb [fab$l_fna]; ! ...
: 294 0385 3         $print_fao ('FAB filename "'!AS"', tmp_desc);
: 295 0386 3     END
: 296 0387 2 ELSE
: 297 0388 2     $print_lit ('FAB filename "" (null)');
: 298 0389 2
: 299 0390 2 IF NOT (.fabb [fab$l_sts])
: 300 0391 2 THEN
: 301 P 0392 2     $print_fao ('FAB$L_STS = !XL    FAB$L_STV = !XL',
: 302 0393 2         .fabb [fab$l_sts], .fabb [fab$l_stv]);
: 303 0394 2
: 304 0395 2 IF nam EQL 0
: 305 0396 2 THEN
: 306 0397 3     BEGIN
: 307 0398 3         $print_lit ('No nam block');
: 308 0399 3         RETURN-1;
: 309 0400 3     END;
: 310 0401 2
: 311 0402 2 IF .nam [nam$b_esl] GTRU 0
: 312 0403 2 THEN
: 313 0404 3     BEGIN
: 314 0405 3         tmp_desc [dsc$w_length] = .nam [nam$b_esl]; ! Create file name desc
: 315 0406 3         tmp_desc [dsc$a_pointer] = .nam [nam$l_esa]; ! ...
: 316 0407 3         $print_fao ('Expanded string "'!AS"', tmp_desc);
: 317 0408 3     END
: 318 0409 2 ELSE
```



```
: 319      0410 2      $print_lit ('Expanded string ''' (null)');
: 320      0411 2
: 321      0412 2      IF .nam [nam$b_rsl] GTRU 0
: 322      0413 2      THEN
: 323      0414 2      BEGIN
: 324      0415 2      tmp_desc [dsc$w_length] = .nam [nam$b_rsl];      ! Create file name desc
: 325      0416 2      tmp_desc [dsc$a_pointer] = .nam [nam$l_rsa];      ! ...
: 326      0417 2      $print_fao ('Result string '''!AS''', tmp_desc);
: 327      0418 2      END;
: 328      0419 2
: 329      0420 2      dev_desc [dsc$w_length] = .nam [nam$b_dev];      ! Create file name desc
: 330      0421 2      dev_desc [dsc$a_pointer] = .nam [nam$l_dev];      ! ...
: 331      0422 2      dir_desc [dsc$w_length] = .nam [nam$b_dir];      ! Create file name desc
: 332      0423 2      dir_desc [dsc$a_pointer] = .nam [nam$l_dir];      ! ...
: 333      0424 2      BEGIN
: 334      0425 2      BIND
: 335      0426 2      len = nam [nam$t_dvi] : BYTE;
: 336      0427 2      dvi_desc [dsc$w_length] = .len;      ! Create file name desc
: 337      0428 2      END;
: 338      0429 2      dvi_desc [dsc$a_pointer] = nam [nam$t_dvi]+1;      ! ...
: 339      0430 2      nam_desc [dsc$w_length] = .nam [nam$b_name];      ! Create file name desc
: 340      0431 2      nam_desc [dsc$a_pointer] = .nam [nam$l_name];      ! ...
: 341      0432 2      nod_desc [dsc$w_length] = .nam [nam$b_node];      ! Create file name desc
: 342      0433 2      nod_desc [dsc$a_pointer] = .nam [nam$l_node];      ! ...
: 343      0434 2      typ_desc [dsc$w_length] = .nam [nam$b_type];      ! Create file name desc
: 344      0435 2      typ_desc [dsc$a_pointer] = .nam [nam$l_type];      ! ...
: 345      0436 2      ver_desc [dsc$w_length] = .nam [nam$b_ver];      ! Create file name desc
: 346      0437 2      ver_desc [dsc$a_pointer] = .nam [nam$l_ver];      ! ...
: 347      0438 2
: 348      P 0439 2      $print_fao ('''!AS'' ''!AS'' ''!AS'' ''!AS'' ''!AS'' ''!AS'' ''!AS''
: 349      0440 2      nod_desc, dev_desc, dir_desc, nam_desc, typ_desc, ver_desc, dvi_desc);
: 350      0441 2
: 351      0442 2      RETURN 1;
: 352      0443 1      END;
```

```
                                .PSECT EXCH$DBG_PLIT,NOWRT,2
21 22 20 65 6D 61 6E 65 6C 69 66 20 42 41 46 006F4 P.ADC: .ASCII \FAB filename ''!AS''\<0><0>
: 320      0411 2
: 321      0412 2      IF .nam [nam$b_rsl] GTRU 0
: 322      0413 2      THEN
: 323      0414 2      BEGIN
: 324      0415 2      tmp_desc [dsc$w_length] = .nam [nam$b_rsl];      ! Create file name desc
: 325      0416 2      tmp_desc [dsc$a_pointer] = .nam [nam$l_rsa];      ! ...
: 326      0417 2      $print_fao ('Result string '''!AS''', tmp_desc);
: 327      0418 2      END;
: 328      0419 2
: 329      0420 2      dev_desc [dsc$w_length] = .nam [nam$b_dev];      ! Create file name desc
: 330      0421 2      dev_desc [dsc$a_pointer] = .nam [nam$l_dev];      ! ...
: 331      0422 2      dir_desc [dsc$w_length] = .nam [nam$b_dir];      ! Create file name desc
: 332      0423 2      dir_desc [dsc$a_pointer] = .nam [nam$l_dir];      ! ...
: 333      0424 2      BEGIN
: 334      0425 2      BIND
: 335      0426 2      len = nam [nam$t_dvi] : BYTE;
: 336      0427 2      dvi_desc [dsc$w_length] = .len;      ! Create file name desc
: 337      0428 2      END;
: 338      0429 2      dvi_desc [dsc$a_pointer] = nam [nam$t_dvi]+1;      ! ...
: 339      0430 2      nam_desc [dsc$w_length] = .nam [nam$b_name];      ! Create file name desc
: 340      0431 2      nam_desc [dsc$a_pointer] = .nam [nam$l_name];      ! ...
: 341      0432 2      nod_desc [dsc$w_length] = .nam [nam$b_node];      ! Create file name desc
: 342      0433 2      nod_desc [dsc$a_pointer] = .nam [nam$l_node];      ! ...
: 343      0434 2      typ_desc [dsc$w_length] = .nam [nam$b_type];      ! Create file name desc
: 344      0435 2      typ_desc [dsc$a_pointer] = .nam [nam$l_type];      ! ...
: 345      0436 2      ver_desc [dsc$w_length] = .nam [nam$b_ver];      ! Create file name desc
: 346      0437 2      ver_desc [dsc$a_pointer] = .nam [nam$l_ver];      ! ...
: 347      0438 2
: 348      P 0439 2      $print_fao ('''!AS'' ''!AS'' ''!AS'' ''!AS'' ''!AS'' ''!AS'' ''!AS''
: 349      0440 2      nod_desc, dev_desc, dir_desc, nam_desc, typ_desc, ver_desc, dvi_desc);
: 350      0441 2
: 351      0442 2      RETURN 1;
: 352      0443 1      END;
```

21	22	20	65	6D	61	6E	65	6C	69	66	20	42	41	46	006F4	P.ADC:	.ASCII \FAB filename ''!AS''\<0><0>
										00	00	22	53	41	00703		
															010E0012	P.ADB:	.LONG 17694738
															00000000		.ADDRESS P.ADC
22	22	20	65	6D	61	6E	65	6C	69	66	20	42	41	46	00710	P.ADE:	.ASCII \FAB filename ''' (null)\<0><0>
						00	00	29	6C	6C	75	6E	28	20	0071F		
															010E0016	P.ADD:	.LONG 17694742
															00000000		.ADDRESS P.ADE
4C	58	21	20	3D	20	53	54	53	5F	4C	24	42	41	46	00730	P.ADG:	.ASCII \FAB\$L_STS = !XL FAB\$L_STV = !XL\<0>
3D	20	56	54	53	5F	4C	24	42	41	46	20	20	20	20	0073F		
										00	4C	58	21	20	0074E		
															00		.ASCII <0>
															010E0022	P.ADF:	.LONG 17694754
															00000000		.ADDRESS P.ADG
			6B	63	6F	6C	62	20	6D	61	6E	20	6F	4E	0075C	P.ADI:	.ASCII \No nam block\
															010E000C	P.ADH:	.LONG 17694732
															00000000		.ADDRESS P.ADI
67	6E	69	72	74	73	20	64	65	64	6E	61	70	78	45	00770	P.ADK:	.ASCII \Expanded string ''!AS''\<0><0><0>







		0B	A3	95	0009C	4\$:	TSTB	11(R3)	:	0402
		18	13	0009F			BEQL	5\$	:	
38	AE	0B	A3	9B	000A1		MOVZBW	11(R3), TMP_DESC	:	0405
3C	AE	0C	A3	D0	000A6		MOVL	12(R3), TMP_DESC+4	:	0406
		38	AE	9F	000AB		PUSHAB	TMP_DESC	:	0407
		0000'	CF	9F	000AE		PUSHAB	P.ADJ	:	
	64		02	FB	000B2		CALLS	#2, EXCH\$UTIL_FAO_BUFFER	:	
			50	DD	000B5		PUSHL	R0	:	
			04	11	000B7		BRB	6\$	:	
		0000'	CF	9F	000B9	5\$:	PUSHAB	P.ADL	:	0410
	65		01	FB	000BD	6\$:	CALLS	#1, LIB\$PUT_OUTPUT	:	
		03	A3	95	000C0		TSTB	3(R3)	:	0412
			19	13	000C3		BEQL	7\$	:	
38	AE	03	A3	9B	000C5		MOVZBW	3(R3), TMP_DESC	:	0415
3C	AE	04	A3	D0	000CA		MOVL	4(R3), TMP_DESC+4	:	0416
		38	AE	9F	000CF		PUSHAB	TMP_DESC	:	0417
		0000'	CF	9F	000D2		PUSHAB	P.ADN	:	
	64		02	FB	000D6		CALLS	#2, EXCH\$UTIL_FAO_BUFFER	:	
			50	DD	000D9		PUSHL	R0	:	
	65		01	FB	000DB		CALLS	#1, LIB\$PUT_OUTPUT	:	
30	AE	39	A3	9B	000DE	7\$:	MOVZBW	57(R3), DEV_DESC	:	0420
34	AE	44	A3	D0	000E3		MOVL	68(R3), DEV_DESC+4	:	0421
28	AE	3A	A3	9B	000E8		MOVZBW	58(R3), DIR_DESC	:	0422
2C	AE	48	A3	D0	000ED		MOVL	72(R3), DIR_DESC+4	:	0423
20	AE	14	A3	9B	000F2		MOVZBW	20(R3), DVI_DESC	:	0427
24	AE	15	A3	9E	000F7		MOVAB	21(R3), DVI_DESC+4	:	0429
18	AE	3B	A3	9B	000FC		MOVZBW	59(R3), NAM_DESC	:	0430
1C	AE	4C	A3	D0	00101		MOVL	76(R3), NAM_DESC+4	:	0431
10	AE	38	A3	9B	00106		MOVZBW	56(R3), NOD_DESC	:	0432
14	AE	40	A3	D0	0010B		MOVL	64(R3), NOD_DESC+4	:	0433
08	AE	3C	A3	9B	00110		MOVZBW	60(R3), TYP_DESC	:	0434
0C	AE	50	A3	D0	00115		MOVL	80(R3), TYP_DESC+4	:	0435
	6E	3D	A3	9B	0011A		MOVZBW	61(R3), VER_DESC	:	0436
04	AE	54	A3	D0	0011E		MOVL	84(R3), VER_DESC+4	:	0437
		20	AE	9F	00123		PUSHAB	DVI_DESC	:	0440
		04	AE	9F	00126		PUSHAB	VER_DESC	:	
		10	AE	9F	00129		PUSHAB	TYP_DESC	:	
		24	AE	9F	0012C		PUSHAB	NAM_DESC	:	
		38	AE	9F	0012F		PUSHAB	DIR_DESC	:	
		44	AE	9F	00132		PUSHAB	DEV_DESC	:	
		28	AE	9F	00135		PUSHAB	NOD_DESC	:	
		0000'	CF	9F	00138		PUSHAB	P.ADP	:	
	64		08	FB	0013C		CALLS	#8, EXCH\$UTIL_FAO_BUFFER	:	
			50	DD	0013F		PUSHL	R0	:	
	65		01	FB	00141	8\$:	CALLS	#1, LIB\$PUT_OUTPUT	:	0442
	50		01	D0	00144		MOVL	#1, R0	:	0443
			04	00147			RET		:	

; Routine Size: 328 bytes, Routine Base: EXCH\$DBG\_CODE + 0386



```
354 0444 1 GLOBAL ROUTINE exch$dbg_dump_rt11_file (filb : $ref_bblock) = %SBTTL 'exch$dbg_dump_rt11_file'
355 0445 2 BEGIN
356 0446 2
357 0447 2 LOCAL
358 0448 2     buf : VECTOR [512, BYTE],
359 0449 2     pbn,
360 0450 2     cnt,
361 0451 2     status;
362 0452 2
363 0453 2 EXTERNAL ROUTINE
364 0454 2     exch$io_rt11_read;
365 0455 2
366 0456 2 BIND
367 0457 2     bufw = buf : VECTOR [256, WORD],
368 0458 2     ctx = filb [filb$a_context] : $ref_bblock,
369 0459 2     volb = filb [filb$a_assoc_volb] : $ref_bblock;
370 0460 2
371 0461 2 $dbgtrc_prefix ('dbg_dump_rt11_file> ');
372 0462 2
373 0463 2 $block_check (2, .filb, filb, 414);
374 0464 2 $block_check (2, .volb, volb, 415);
375 0465 2 $block_check (2, .ctx, rt11ctx, 416);
376 0466 2 $logic_check (2, (.ctx [rt11ctx$a_assoc_filb] EQL .filb), 106);
377 0467 2 $logic_check (2, (.ctx [rt11ctx$a_assoc_volb] EQL .volb), 107);
378 0468 2
379 0469 2 pbn = .ctx [rt11ctx$l_start_block];
380 0470 2 cnt = .ctx [rt11ctx$w_blocks];
381 0471 2
382 0472 2 INCRU blk FROM .pbn TO .pbn + .cnt - 1
383 0473 2 DO
384 0474 2     BEGIN
385 0475 2
386 0476 2     IF NOT (status = exch$io_rt11_read (.volb, .blk, 1, buf))
387 0477 2     THEN
388 0478 2         $exch_signal_return (.status);
389 0479 2
390 0480 2     $print_fao ('> !XW Vbn    (!XW Pbn)', .blk-.pbn, .blk);
391 0481 2
392 0482 2     INCRU i FROM 0 TO 15
393 0483 2     DO
394 0484 2         BEGIN
395 0485 2             REGISTER off, offw;
396 0486 2
397 0487 2             off = .i * 32;          offw = .i * 16;
398 0488 2
399 0489 2             $print_fao ('>    !16(5XW) - !XW - '!AF''',
400 P 0490 2                 .bufw [.offw + 15], .bufw [.offw + 14], .bufw [.offw + 13], .bufw [.offw + 12],
401 P 0491 2                 .bufw [.offw + 11], .bufw [.offw + 10], .bufw [.offw + 9], .bufw [.offw + 8],
402 P 0492 2                 .bufw [.offw + 7], .bufw [.offw + 6], .bufw [.offw + 5], .bufw [.offw + 4],
403 P 0493 2                 .bufw [.offw + 3], .bufw [.offw + 2], .bufw [.offw + 1], .bufw [.offw],
404 0494 2                 .off, 32, buf [.off]);
405 0495 2
406 0496 2         END;
407 0497 2     END;
408 0498 2
409 0499 2 RETURN 1;
410 0500 1 END;
```

69  
7465  
65

69

69



```
.PSECT EXCH$DBG_PLIT,NOWRT,2
58 21 28 20 20 20 6E 62 56 20 57 58 21 20 3E 00808 P.ADS: .ASCII \> !XW Vbn (!XW Pbn)\<0><0><0>
      00 00 00 29 6E 62 50 20 57 00817
      010E0015 00820 P.ADR: .LONG 17694741
      00000000 00824 .ADDRESS P.ADS
2D 20 29 57 58 35 28 36 31 21 20 20 20 20 3E 00828 P.ADU: .ASCII \> !16(5XW) - !XW - '!AF'\<0>
      00 22 46 41 21 22 20 2D 20 57 58 21 20 00837
      010E001B 00844 P.ADT: .LONG 17694747
      00000000 00848 .ADDRESS P.ADU

.EXTRN EXCH$IO_RT11_READ
.EXTRN EXCH$_BADLOGIC

.PSECT EXCH$DBG_CODE,NOWRT,2
      OFFC 00000
      .ENTRY EXCH$DBG_DUMP_RT11_FILE, Save R2,R3,R4,R5,- 0444
      MOVAB EXCH$UTIL_FAO_BUFFER, R1
      MOVAB LIB$STOP, R10
      MOVL #EXCH$_BADLOGIC, R9
      MOVAB EXCH$UTIL_BLOCK_CHECK, R8
      MOVAB -512(SP), SP
      ADDL3 #32, FILB, R3 0458
      ADDL3 #28, FILB, R6 0459
      MOVL #56295674, R2 0463
      MOVZWL #414, R1
      MOVL FILB, R0
      JSB EXCH$UTIL_BLOCK_CHECK
      MOVL #68878579, R2 0464
      MOVZWL #415, R1
      MOVL (R6), R0
      JSB EXCH$UTIL_BLOCK_CHECK
      MOVL (R3), R4 0465
      MOVL #8519924, R2
      MOVZWL #416, R1
      MOVL R4, R0
      JSB EXCH$UTIL_BLOCK_CHECK
      CMPL 16(R4), FILB 0466
      BEQL 1$
      MOVZBL #106, -(SP)
      PUSHL #1
      PUSHL R9
      CALLS #3, LIB$STOP
      CMPL 20(R4), (R6) 0467
      BEQL 2$
      MOVZBL #107, -(SP)
      PUSHL #1
      PUSHL R9
      CALLS #3, LIB$STOP
      MOVL 114(R4), PBN
      MOVZWL 64(R4), CNT
      MOVAB -1(CNT)[PBN], R5
      MOVL PBN, BLK

5B 00000000G EF 9E 00002
5A 00000000G 00 9E 00009
59 00000000G 8F D0 00010
58 00000000G EF 9E 00017
5E FE00 CE 9E 0001E
AC 20 C1 00023
AC 1C C1 00028
52 035B00FA 8F D0 0002D
51 019E 8F 3C 00034
50 04 AC D0 00039
      68 16 0003D
52 041B00F3 8F D0 0003F
51 019F 8F 3C 00046
50 66 D0 0004B
      68 16 0004E
54 63 D0 00050
52 008200F4 8F D0 00053
51 01A0 8F 3C 0005A
50 54 D0 0005F
      68 16 00062
04 AC 10 A4 D1 00064
      0B 13 00069
7E 6A 8F 9A 0006B
      01 DD 0006F
      59 DD 00071
6A 03 FB 00073
66 14 A4 D1 00076 1$:
      0B 13 0007A
7E 6B 8F 9A 0007C
      01 DD 00080
      59 DD 00082
6A 03 FB 00084
53 72 A4 D0 00087 2$:
50 40 A4 3C 0008B
55 FF A043 9E 0008F
52 53 D0 00094
```



		00B5	31	00097	BRW	6\$		
		5E	DD	0009A	3\$: PUSHL	SP		0476
		01	DD	0009C	PUSHL	#1		
		52	DD	0009E	PUSHL	BLK		
		66	DD	000A0	PUSHL	(R6)		
	00000000G	EF	04	FB	000A2	CALLS	#4, EXCH\$IO_RT11_READ	
		57	50	DD	000A9	MOVL	R0, STATUS	
		10	57	E8	000AC	BLBS	STATUS, 4\$	
		54	57	DD	000AF	MOVL	STATUS, TEMP	0478
			54	DD	000B2	PUSHL	TEMP	
	00000000G	00	01	FB	000B4	CALLS	#1, LIB\$SIGNAL	
		50	54	DD	000BB	MOVL	TEMP, R0	
				04	000BE	RET		
			52	DD	000BF	4\$: PUSHL	BLK	0480
7E		52	53	C3	000C1	SUBL3	PBN, BLK, -(SP)	
			CF	9F	000C5	PUSHAB	P.ADR	
		6B	03	FB	000C9	CALLS	#3, EXCH\$UTIL_FAO_BUFFER	
			50	DD	000CC	PUSHL	R0	
	00000000G	00	01	FB	000CE	CALLS	#1, LIB\$PUT_OUTPUT	
			54	D4	000D5	CLRL	I	0482
50		54	05	78	000D7	5\$: ASHL	#5, I, OFF	0487
51		54	04	78	000DB	ASHL	#4, I, OFFW	
			6E40	9F	000DF	PUSHAB	BUF[OFF]	0494
			20	DD	000E2	PUSHL	#32	
			50	DD	000E4	PUSHL	OFF	
		7E	0C	AE41	3C	000E6	MOVZWL	BUFW[OFFW], -(SP)
		7E	12	AE41	3C	000EB	MOVZWL	BUFW+2[OFFW], -(SP)
		7E	18	AE41	3C	000F0	MOVZWL	BUFW+4[OFFW], -(SP)
		7E	1E	AE41	3C	000F5	MOVZWL	BUFW+6[OFFW], -(SP)
		7E	24	AE41	3C	000FA	MOVZWL	BUFW+8[OFFW], -(SP)
		7E	2A	AE41	3C	000FF	MOVZWL	BUFW+10[OFFW], -(SP)
		7E	30	AE41	3C	00104	MOVZWL	BUFW+12[OFFW], -(SP)
		7E	36	AE41	3C	00109	MOVZWL	BUFW+14[OFFW], -(SP)
		7E	3C	AE41	3C	0010E	MOVZWL	BUFW+16[OFFW], -(SP)
		7E	42	AE41	3C	00113	MOVZWL	BUFW+18[OFFW], -(SP)
		7E	48	AE41	3C	00118	MOVZWL	BUFW+20[OFFW], -(SP)
		7E	4E	AE41	3C	0011D	MOVZWL	BUFW+22[OFFW], -(SP)
		7E	54	AE41	3C	00122	MOVZWL	BUFW+24[OFFW], -(SP)
		7E	5A	AE41	3C	00127	MOVZWL	BUFW+26[OFFW], -(SP)
		7E	60	AE41	3C	0012C	MOVZWL	BUFW+28[OFFW], -(SP)
		7E	66	AE41	3C	00131	MOVZWL	BUFW+30[OFFW], -(SP)
			0000	CF	9F	00136	PUSHAB	P.ADT
		6B	14	FB	0013A	CALLS	#20, EXCH\$UTIL_FAO_BUFFER	
			50	DD	0013D	PUSHL	R0	
	00000000G	00	01	FB	0013F	CALLS	#1, LIB\$PUT_OUTPUT	
			54	D6	00146	INCL	I	0482
		0F	54	D1	00148	CMPL	I, #15	
			8A	1B	0014B	BLEQU	5\$	
			52	D6	0014D	INCL	BLK	0472
		55	52	D1	0014F	6\$: CMPL	BLK, R5	
			03	1A	00152	BGTRU	7\$	
			FF43	31	00154	BRW	3\$	
		50	01	DD	00157	7\$: MOVL	#1, R0	0499
				04	0015A	RET		0500

; Routine Size: 347 bytes, Routine Base: EXCH\$DBG\_CODE + 04CE

EXCH\$DBG  
V04-000

Conditionally assembled debugging routines  
exch\$dbg\_dump\_rt11\_file

K 6  
16-Sep-1984 00:45:26  
14-Sep-1984 12:29:02

VAX-11 Bliss-32 V4.0-742  
[EXCHNG.SRC]EXCDBG.B32;1

Page 24  
(7)

EXCI  
V04.

69  
6E

68  
6E

68  
69



```
412 0501 1 GLOBAL ROUTINE exch$dbg_nam_dump (nam : $ref_bblock) = %SBTTL 'exch$dbg_nam_dump'
413 0502 2 BEGIN
414 0503 2
415 0504 2 $dbgtrc_prefix ('nam_dump> ');
416 0505 2
417 0506 2 !
418 0507 2 ! A macro to add successive ASCII items to a string buffer, and update the
419 0508 2 ! descriptor for the string.
420 0509 2 !
421 0510 2 MACRO
422 0511 2     add_to_list (string) =
423 0512 2     BEGIN
424 0513 2         BIND desc = %ASCID string : $BBLOCK;
425 0514 2         CH$MOVE(.desc[dsc$w_length],
426 0515 2             .desc[dsc$a_pointer],
427 0516 2             .ptr[1] + .ptr[0]);
428 0517 2         ptr[0] = .ptr[0] + .desc[dsc$w_length];
429 0518 2     END%;
430 0519 2
431 0520 2 LOCAL
432 0521 2     buffer : VECTOR [255, BYTE],
433 0522 2     ptr : VECTOR [2, LONG];
434 0523 2
435 0524 2     $print_fao ('      Dump of nam at !XL:', .nam);
436 0525 2     $print_fao ('      !15AS !XB - block id ', %ASCID 'NAM$B BID', .nam[NAM$B BID]);
437 0526 2     $print_fao ('      !15AS !XB - block length ', %ASCID 'NAM$B BLN', .nam[NAM$B BLN]);
438 0527 2     $print_fao ('      !15AS !XB - resultant string area size ', %ASCID 'NAM$B RSS', .nam[NAM$B RSS]);
439 0528 2     $print_fao ('      !15AS !XB - name options ', %ASCID 'NAM$B NOP', .nam[NAM$B NOP]);
440 0529 2     $print_fao ('      !15AS !XL - wild card context ', %ASCID 'NAM$B WCC', .nam[NAM$B WCC]);
441 0530 2     ptr[0] = 0;
442 0531 2     ptr[1] = buffer;
443 0532 2     IF .nam[nam$l_fnb] NEQ 0
444 0533 2     THEN
445 0534 2     BEGIN
446 0535 2     IF .nam[NAM$V_EXP_VER] THEN add_to_list('xver ');
447 0536 2     IF .nam[NAM$V_EXP_TYPE] THEN add_to_list('xtyp ');
448 0537 2     IF .nam[NAM$V_EXP_NAME] THEN add_to_list('xnam ');
449 0538 2     IF .nam[NAM$V_WILD_VER] THEN add_to_list('*ver ');
450 0539 2     IF .nam[NAM$V_WILD_TYPE] THEN add_to_list('*typ ');
451 0540 2     IF .nam[NAM$V_WILD_NAME] THEN add_to_list('*nam ');
452 0541 2     IF .nam[NAM$V_EXP_DIR] THEN add_to_list('xdir ');
453 0542 2     IF .nam[NAM$V_EXP_DEV] THEN add_to_list('xdev ');
454 0543 2     IF .nam[NAM$V_WILD_CARD] THEN add_to_list('*** ');
455 0544 2     IF .nam[NAM$V_SEARCH_LIST] THEN add_to_list('schl ');
456 0545 2     IF .nam[NAM$V_CNCL_DEV] THEN add_to_list('cncl ');
457 0546 2     IF .nam[NAM$V_ROOT_DIR] THEN add_to_list('root ');
458 0547 2     IF .nam[NAM$V_LOWVER] THEN add_to_list('lowv ');
459 0548 2     IF .nam[NAM$V_HIGHVER] THEN add_to_list('hiv ');
460 0549 2     IF .nam[NAM$V_PPF] THEN add_to_list('ppf ');
461 0550 2     IF .nam[NAM$V_NODE] THEN add_to_list('nod ');
462 0551 2     IF .nam[NAM$V_QUOTED] THEN add_to_list('quot ');
463 0552 2     IF .nam[NAM$V_WILD_DIR] THEN add_to_list('*dir ');
464 0553 2     IF .nam[NAM$V_GRP_MBR] THEN BEGIN
465 0554 2     add_to_list('grpmem ');
466 0555 2     IF .nam[NAM$V_WILD_GRP] THEN add_to_list('*grp ');
467 0556 2     IF .nam[NAM$V_WILD_MBR] THEN add_to_list('*mbr ');
468 0557 2     END;
```



```

469      0558      3      IF .ptr[0] GTR 0 THEN ptr[0] = .ptr[0] - 1;      ! Remove final space
470      0559      2      END;
471      0560      2      $print_fao ('      !15AS      !XL      - name bits "'AS'", %ASCID 'NAM$FNB', .nam[NAM$FNB], ptr);
472      P 0561      2      $print_fao ('      !15AS      !XB !XL      "'AF'", %ASCID 'expanded string',
473      0562      2      .nam[NAM$ESL], .nam[NAM$ESA], .nam[nam$b_esl], .nam[nam$l_esa]);
474      P 0563      2      $print_fao ('      !15AS      !XB !XL      "'AF'", %ASCID 'result string',
475      0564      2      .nam[NAM$RSL], .nam[NAM$RSA], .nam[nam$b_rsl], .nam[nam$l_rsa]);
476      0565      2      $print_fao ('      !15AS      "'AC" - device name', %ASCID 'NAM$I_DVI', nam[NAM$I_DVI]);
477      P 0566      2      $print_fao ('      !AF" "'AF" "'AF" "'AF" "'AF" "'AF"
478      P 0567      2      .nam[NAM$B_NODE], .nam[NAM$L_NODE],
479      P 0568      2      .nam[NAM$B_DEV], .nam[NAM$L_DEV],
480      P 0569      2      .nam[NAM$B_DIR], .nam[NAM$L_DIR],
481      P 0570      2      .nam[NAM$B_NAME], .nam[NAM$L_NAME],
482      P 0571      2      .nam[NAM$B_TYPE], .nam[NAM$L_TYPE],
483      0572      2      .nam[NAM$B_VER], .nam[NAM$L_VER]);
484      0573      2      $print_fao ('      !15AS      !XL      - related file nam block addr ', %ASCID 'NAM$L_RLF', .nam[NAM$L_
485      0574      2      IF .nam[NAM$L_RLF] NEQ 0
486      0575      2      THEN exch$dbg_nam_dump (.nam[NAM$L_RLF]);
487      0576      2
488      0577      2      RETURN 1;
489      0578      1      END;

```

														.PSECT	EXCH\$DBG_PLIT,NOWRT,2	
6D	61	6E	20	66	6F	20	70	6D	75	44	20	20	20	0084C	P.ADW:	.ASCII \ Dump of nam at !XL:\<0>
						00	3A	4C	58	21	20	74	61	0085B		
														010E0017,	00864	P.ADV: .LONG 17694743
														00000000'	00868	.ADDRESS P.ADW
20	20	53	41	35	31	21	20	20	20	20	20	20	20	0086C	P.ADY:	.ASCII \ !15AS !XB - name o\
20	20	20	20	20	20	20	20	42	58	21	20	20	20	0087B		
					6F	20	65	6D	61	6E	20	2D	20	0088A		
							00	20	73	6E	6F	69	74	00894		
														010E002F,	0089C	P.ADX: .ASCII \ptions \<0>
														00000000'	008AO	.LONG 17694767
																.ADDRESS P.ADY
				00	00	00	50	4F	4E	5F	42	24	4D	008A4	P.AEA:	.ASCII \NAMSB NOP\<0><0><0>
														010E0009,	008BO	P.ADZ: .LONG 17694729
														00000000'	008B4	.ADDRESS P.AEA
20	20	53	41	35	31	21	20	20	20	20	20	20	20	008BB	P.AEC:	.ASCII \ !15AS !XL - wild card co\
69	77	20	2D	20	20	20	20	4C	58	21	20	20	20	008C7		
					6F	63	20	64	72	61	63	20	64	008D6		
							00	20	20	74	78	65	74	008EO		
														010E002F,	008EB	P.AEB: .ASCII \ntext \<0>
														00000000'	008EC	.LONG 17694767
																.ADDRESS P.AEC
				00	00	00	43	43	57	5F	4C	24	4D	008FO	P.AEE:	.ASCII \NAMS L WCC\<0><0><0>
														010E0009,	008FC	P.AED: .LONG 17694729
														00000000'	00900	.ADDRESS P.AEE
							00	00	00	20	72	65	76	00904	P.AEG:	.ASCII \xver \<0><0><0>
														010E0005,	0090C	P.AEF: .LONG 17694725
														00000000'	00910	.ADDRESS P.AEG
							00	00	00	20	70	79	74	00914	P.AEI:	.ASCII \xtyp \<0><0><0>
														010E0005,	0091C	P.AEH: .LONG 17694725
														00000000'	00920	.ADDRESS P.AEI
							00	00	00	20	6D	61	6E	00924	P.AEK:	.ASCII \xnam \<0><0><0>
														010E0005,	0092C	P.AEJ: .LONG 17694725
														00000000'	00930	.ADDRESS P.AEK



```
00 00 00 20 72 65 76 2A 00934 P.AEM: .ASCII \*ver \<0><0><0>
010E0005 0093C P.AEL: .LONG 17694725
00000000 00940 .ADDRESS P.AEM
00 00 00 20 70 79 74 2A 00944 P.AEO: .ASCII \*typ \<0><0><0>
010E0005 0094C P.AEN: .LONG 17694725
00000000 00950 .ADDRESS P.AEO
00 00 00 20 6D 61 6E 2A 00954 P.AEQ: .ASCII \*nam \<0><0><0>
010E0005 0095C P.AEP: .LONG 17694725
00000000 00960 .ADDRESS P.AEQ
00 00 00 20 72 69 64 78 00964 P.AES: .ASCII \xdir \<0><0><0>
010E0005 0096C P.AER: .LONG 17694725
00000000 00970 .ADDRESS P.AES
00 00 00 20 76 65 64 78 00974 P.AEU: .ASCII \xdev \<0><0><0>
010E0005 0097C P.AET: .LONG 17694725
00000000 00980 .ADDRESS P.AEU
20 2A 2A 2A 00984 P.AEW: .ASCII \*** \
010E0004 00988 P.AEV: .LONG 17694724
00000000 0098C .ADDRESS P.AEW
00 00 00 20 6C 68 63 73 00990 P.AEY: .ASCII \schl \<0><0><0>
010E0005 00998 P.AEX: .LONG 17694725
00000000 0099C .ADDRESS P.AEY
00 00 00 20 6C 63 6E 63 009A0 P.AFA: .ASCII \cncl \<0><0><0>
010E0005 009A8 P.AEZ: .LONG 17694725
00000000 009AC .ADDRESS P.AFA
00 00 00 20 74 6F 6F 72 009B0 P.AFC: .ASCII \root \<0><0><0>
010E0005 009B8 P.AFB: .LONG 17694725
00000000 009BC .ADDRESS P.AFC
00 00 00 20 76 77 6F 6C 009C0 P.AFE: .ASCII \lowv \<0><0><0>
010E0005 009C8 P.AFD: .LONG 17694725
00000000 009CC .ADDRESS P.AFE
20 76 69 68 009D0 P.AFG: .ASCII \hiv \
010E0004 009D4 P.AFF: .LONG 17694724
00000000 009D8 .ADDRESS P.AFG
20 66 70 70 009DC P.AFI: .ASCII \ppf \
010E0004 009E0 P.AFH: .LONG 17694724
00000000 009E4 .ADDRESS P.AFI
20 64 6F 6E 009E8 P.AFK: .ASCII \nod \
010E0004 009EC P.AFJ: .LONG 17694724
00000000 009F0 .ADDRESS P.AFK
00 00 00 20 74 6F 75 71 009F4 P.AFM: .ASCII \quot \<0><0><0>
010E0005 009FC P.AFL: .LONG 17694725
00000000 00A00 .ADDRESS P.AFM
00 00 00 20 72 69 64 2A 00A04 P.AFO: .ASCII \*dir \<0><0><0>
010E0005 00A0C P.AFN: .LONG 17694725
00000000 00A10 .ADDRESS P.AFO
00 20 6D 65 6D 70 72 67 00A14 P.AFQ: .ASCII \grpmem \<0>
010E0007 00A1C P.AFP: .LONG 17694727
00000000 00A20 .ADDRESS P.AFQ
00 00 00 20 70 72 67 2A 00A24 P.AFS: .ASCII \*grp \<0><0><0>
010E0005 00A2C P.AFR: .LONG 17694725
00000000 00A30 .ADDRESS P.AFS
00 00 00 20 72 62 6D 2A 00A34 P.AFU: .ASCII \*mbr \<0><0><0>
010E0005 00A3C P.AFT: .LONG 17694725
00000000 00A40 .ADDRESS P.AFU
20 20 53 41 35 31 21 20 20 20 20 20 20 20 20 00A44 P.AFW: .ASCII \ !15AS !XL - name bits ""\
61 6E 20 2D 20 20 20 20 4C 58 21 20 20 20 20 00A53
21 22 20 73 74 69 62 20 65 6D 00A62
```



```
00 22 53 41 00A6C .ASCII \AS'\<0>
010E002B 00A70 P.AFV: .LONG 17694763
00000000 00A74 .ADDRESS P.AFW
00 00 00 42 4E 46 5F 4C 24 4D 41 4E 00A78 P.AFY: .ASCII \NAMSL FNB\<0><0><0>
010E0009 00A84 P.AFX: .LONG 17694729
00000000 00A88 .ADDRESS P.AFY
20 20 53 41 35 31 21 20 20 20 20 20 20 20 20 00A8C P.AGA: .ASCII \ !15AS !XB !XL '!AF'\
41 21 22 20 20 20 20 4C 58 21 20 42 58 21 20 00A9B
22 46 00AAA
010E0020 00AAC P.AFZ: .LONG 17694752
00000000 00AB0 .ADDRESS P.AGA
67 6E 69 72 74 73 20 64 65 64 6E 61 70 78 65 00AB4 P.AGC: .ASCII \expanded string\<0>
00 00AC3
010E000F 00AC4 P.AGB: .LONG 17694735
00000000 00AC8 .ADDRESS P.AGC
20 20 53 41 35 31 21 20 20 20 20 20 20 20 20 00ACC P.AGE: .ASCII \ !15AS !XB !XL '!AF'\
41 21 22 20 20 20 20 4C 58 21 20 42 58 21 20 00ADB
22 46 00AEA
010E0020 00AEC P.AGD: .LONG 17694752
00000000 00AF0 .ADDRESS P.AGE
00 00 67 6E 69 72 74 73 20 74 6C 75 73 65 72 00AF4 P.AGG: .ASCII \result string\<0><0><0>
00 00B03
010E000D 00B04 P.AGF: .LONG 17694733
00000000 00B08 .ADDRESS P.AGG
20 20 53 41 35 31 21 20 20 20 20 20 20 20 20 00B0C P.AGI: .ASCII \ !15AS '!AC' - device name\
64 20 20 20 20 20 20 22 43 41 21 22 20 20 20 00B1B
65 6D 61 6E 20 65 63 69 76 65 00B2A
010E0028 00B34 P.AGH: .LONG 17694760
00000000 00B38 .ADDRESS P.AGI
00 00 00 49 56 44 5F 54 24 4D 41 4E 00B3C P.AGK: .ASCII \NAMSL DVI\<0><0><0>
010E0009 00B48 P.AGJ: .LONG 17694729
00000000 00B4C .ADDRESS P.AGK
22 20 22 46 41 21 22 20 20 20 20 20 20 20 20 00B50 P.AGM: .ASCII \ '!AF' '!AF' '!AF' '!AF' '!AF' '!AF' '!AF'
46 41 21 22 20 22 46 41 21 22 20 22 46 41 21 00B5F
21 22 20 22 46 41 21 22 20 22 46 41 21 22 20 22 00B6E
00 22 46 41 00B78
010E002B 00B7C P.AGL: .ASCII \AF'\<0>
00000000 00B80 .LONG 17694763
20 20 53 41 35 31 21 20 20 20 20 20 20 20 20 00B84 P.AGO: .ADDRESS P.AGM
65 72 20 20 20 20 20 20 4C 58 21 20 20 20 20 00B93
72 64 64 61 20 6B 63 6F 6C 62 20 6D 61 6E 20 00BA2
00 00 00 20 20 00BAC .ASCII \ nam block addr \<0><0><0>
010E0039 00BC0 P.AGN: .LONG 17694777
00000000 00BC4 .ADDRESS P.AGO
00 00 00 46 4C 52 5F 4C 24 4D 41 4E 00BC8 P.AGQ: .ASCII \NAMSL RLF\<0><0><0>
010E0009 00BD4 P.AGP: .LONG 17694729
00000000 00BD8 .ADDRESS P.AGQ
```

```
DESC= P.AEF
DESC= P.AEH
DESC= P.AEJ
DESC= P.AEL
DESC= P.AEN
DESC= P.AEP
DESC= P.AER
DESC= P.AET
```



DESC=	P.AEV
DESC=	P.AEX
DESC=	P.AEZ
DESC=	P.AFB
DESC=	P.AFD
DESC=	P.AFF
DESC=	P.AFH
DESC=	P.AFJ
DESC=	P.AFL
DESC=	P.AFN
DESC=	P.AFP
DESC=	P.AFR
DESC=	P.AFT

.ENTRY	EXCH\$DBG NAM_DUMP, Save R2,R3,R4,R5,R6,R7,-	0501
	R8,R9,R10	:
MOVAB	LIB\$PUT_OUTPUT, R10	:
MOVAB	EXCH\$UTIL_FAO_BUFFER, R9	:
MOVAB	DESC, R8	:
MOVAB	-264(SP), SP	:
MOVL	NAM, R6	0524
PUSHL	R6	:
PUSHAB	P.ADV	:
CALLS	#2, EXCH\$UTIL_FAO_BUFFER	:
PUSHL	R0	:
CALLS	#1, LIB\$PUT_OUTPUT	:
MOVZBL	8(R6), -(SP)	0528
PUSHAB	P.ADZ	:
PUSHAB	P.ADX	:
CALLS	#3, EXCH\$UTIL_FAO_BUFFER	:
PUSHL	R0	:
CALLS	#1, LIB\$PUT_OUTPUT	:
PUSHL	48(R6)	0529
PUSHAB	P.AED	:
PUSHAB	P.AEB	:
CALLS	#3, EXCH\$UTIL_FAO_BUFFER	:
PUSHL	R0	:
CALLS	#1, LIB\$PUT_OUTPUT	:
CLRL	PTR	0530
MOVAB	BUFFER, PTR+4	0531
MOVAB	52(R6), R7	0532
TSTL	(R7)	:
BNEQ	1\$	:
BRW	22\$	:
BLBC	(R7), 2\$	0535
ADDL3	PTR, PTR+4, R0	:
MOVC3	DESC, @DESC+4, (R0)	:
MOVZWL	DESC, R0	:
ADDL2	R0, PTR	:
BBC	#1, (R7), 3\$	0536
ADDL3	PTR, PTR+4, R0	:
MOVC3	DESC, @DESC+4, (R0)	:
MOVZWL	DESC, R0	:
ADDL2	R0, PTR	:



12		67	02	E1	0008A	3\$:	BBC	#2, (R7), 4\$	: 0537	
50	04	AE	6E	C1	0008E		ADDL3	PTR, PTR+4, R0	:	
60	24	B8	20	A8	28	00093	MOVC3	DESC, @DESC+4, (R0)	:	
		50	20	A8	3C	00099	MOVZWL	DESC, R0	:	
		6E		50	C0	0009D	ADDL2	R0, PTR	:	
12		67	03	E1	000A0	4\$:	BBC	#3, (R7), 5\$	: 0538	
50	04	AE	6E	C1	000A4		ADDL3	PTR, PTR+4, R0	:	
60	34	B8	30	A8	28	000A9	MOVC3	DESC, @DESC+4, (R0)	:	
		50	30	A8	3C	000AF	MOVZWL	DESC, R0	:	
		6E		50	C0	000B3	ADDL2	R0, PTR	:	
12		67	04	E1	000B6	5\$:	BBC	#4, (R7), 6\$	: 0539	
50	04	AE	6E	C1	000BA		ADDL3	PTR, PTR+4, R0	:	
60	44	B8	40	A8	28	000BF	MOVC3	DESC, @DESC+4, (R0)	:	
		50	40	A8	3C	000C5	MOVZWL	DESC, R0	:	
		6E		50	C0	000C9	ADDL2	R0, PTR	:	
12		67	05	E1	000CC	6\$:	BBC	#5, (R7), 7\$	: 0540	
50	04	AE	6E	C1	000D0		ADDL3	PTR, PTR+4, R0	:	
60	54	B8	50	A8	28	000D5	MOVC3	DESC, @DESC+4, (R0)	:	
		50	50	A8	3C	000DB	MOVZWL	DESC, R0	:	
		6E		50	C0	000DF	ADDL2	R0, PTR	:	
12		67	06	E1	000E2	7\$:	BBC	#6, (R7), 8\$	: 0541	
50	04	AE	6E	C1	000E6		ADDL3	PTR, PTR+4, R0	:	
60	64	B8	60	A8	28	000EB	MOVC3	DESC, @DESC+4, (R0)	:	
		50	60	A8	3C	000F1	MOVZWL	DESC, R0	:	
		6E		50	C0	000F5	ADDL2	R0, PTR	:	
				67	95	000F8	8\$:	TSTB	(R7)	: 0542
				12	18	000FA	BGEQ	9\$		:
50	04	AE	6E	C1	000FC		ADDL3	PTR, PTR+4, R0	:	
60	74	B8	70	A8	28	00101	MOVC3	DESC, @DESC+4, (R0)	:	
		50	70	A8	3C	00107	MOVZWL	DESC, R0	:	
		6E		50	C0	0010B	ADDL2	R0, PTR	:	
		13	01	A7	E9	0010E	9\$:	BLBC	1(R7), 10\$	: 0543
50	04	AE	6E	C1	00112		ADDL3	PTR, PTR+4, R0	:	
60	0080	D8	7C	A8	28	00117	MOVC3	DESC, @DESC+4, (R0)	:	
		50	7C	A8	3C	0011E	MOVZWL	DESC, R0	:	
		6E		50	C0	00122	ADDL2	R0, PTR	:	
15		67	0B	E1	00125	10\$:	BBC	#11, (R7), 11\$	: 0544	
50	04	AE	6E	C1	00129		ADDL3	PTR, PTR+4, R0	:	
60	0090	D8	008C	C8	28	0012E	MOVC3	DESC, @DESC+4, (R0)	:	
		50	008C	C8	3C	00136	MOVZWL	DESC, R0	:	
		6E		50	C0	0013B	ADDL2	R0, PTR	:	
15		67	0C	E1	0013E	11\$:	BBC	#12, (R7), 12\$	: 0545	
50	04	AE	6E	C1	00142		ADDL3	PTR, PTR+4, R0	:	
60	00A0	D8	009C	C8	28	00147	MOVC3	DESC, @DESC+4, (R0)	:	
		50	009C	C8	3C	0014F	MOVZWL	DESC, R0	:	
		6E		50	C0	00154	ADDL2	R0, PTR	:	
15		67	0D	E1	00157	12\$:	BBC	#13, (R7), 13\$	: 0546	
50	04	AE	6E	C1	0015B		ADDL3	PTR, PTR+4, R0	:	
60	00B0	D8	00AC	C8	28	00160	MOVC3	DESC, @DESC+4, (R0)	:	
		50	00AC	C8	3C	00168	MOVZWL	DESC, R0	:	
		6E		50	C0	0016D	ADDL2	R0, PTR	:	
15		67	0E	E1	00170	13\$:	BBC	#14, (R7), 14\$	: 0547	
50	04	AE	6E	C1	00174		ADDL3	PTR, PTR+4, R0	:	
60	00C0	D8	00BC	C8	28	00179	MOVC3	DESC, @DESC+4, (R0)	:	
		50	00BC	C8	3C	00181	MOVZWL	DESC, R0	:	
		6E		50	C0	00186	ADDL2	R0, PTR	:	
				67	B5	00189	14\$:	TSTW	(R7)	: 0548



50	04	AE	15	18	0018B	BGEQ	15\$		
60	00CC	D8	6E	C1	0018D	ADDL3	PTR, PTR+4, R0		
		50	00C8	C8	28	00192	MOVC3	DESC, @DESC+4, (R0)	
		6E	00C8	C8	3C	0019A	MOVZWL	DESC, R0	
		15	02	50	C0	0019F	ADDL2	R0, PTR	
50	04	AE	A7	E9	001A2	15\$: BLBC	2(R7), 16\$		0549
60	00D8	D8	6E	C1	001A6	ADDL3	PTR, PTR+4, R0		
		50	00D4	C8	28	001AB	MOVC3	DESC, @DESC+4, (R0)	
		6E	00D4	C8	3C	001B3	MOVZWL	DESC, R0	
15		67	50	C0	001B8	ADDL2	R0, PTR		
50	04	AE	11	E1	001BB	16\$: BBC	#17, (R7), 17\$		0550
60	00E4	D8	6E	C1	001BF	ADDL3	PTR, PTR+4, R0		
		50	00E0	C8	28	001C4	MOVC3	DESC, @DESC+4, (R0)	
		6E	00E0	C8	3C	001CC	MOVZWL	DESC, R0	
15		67	50	C0	001D1	ADDL2	R0, PTR		
50	04	AE	12	E1	001D4	17\$: BBC	#18, (R7), 18\$		0551
60	00F4	D8	6E	C1	001D8	ADDL3	PTR, PTR+4, R0		
		50	00F0	C8	28	001DD	MOVC3	DESC, @DESC+4, (R0)	
		6E	00F0	C8	3C	001E5	MOVZWL	DESC, R0	
15		67	50	C0	001EA	ADDL2	R0, PTR		
50	04	AE	14	E1	001ED	18\$: BBC	#20, (R7), 19\$		0552
60	0104	D8	6E	C1	001F1	ADDL3	PTR, PTR+4, R0		
		50	0100	C8	28	001F6	MOVC3	DESC, @DESC+4, (R0)	
		6E	0100	C8	3C	001FE	MOVZWL	DESC, R0	
47		67	50	C0	00203	ADDL2	R0, PTR		
50	04	AE	13	E1	00206	19\$: BBC	#19, (R7), 21\$		0553
60	0114	D8	6E	C1	0020A	ADDL3	PTR, PTR+4, R0		0554
		50	0110	C8	28	0020F	MOVC3	DESC, @DESC+4, (R0)	
		6E	0110	C8	3C	00217	MOVZWL	DESC, R0	
		15	03	50	C0	0021C	ADDL2	R0, PTR	
50	04	AE	A7	E9	0021F	BLBC	3(R7), 20\$		0555
60	0124	D8	6E	C1	00223	ADDL3	PTR, PTR+4, R0		
		50	0120	C8	28	00228	MOVC3	DESC, @DESC+4, (R0)	
		6E	0120	C8	3C	00230	MOVZWL	DESC, R0	
15		67	50	C0	00235	ADDL2	R0, PTR		
50	04	AE	19	E1	00238	20\$: BBC	#25, (R7), 21\$		0556
60	0134	D8	6E	C1	0023C	ADDL3	PTR, PTR+4, R0		
		50	0130	C8	28	00241	MOVC3	DESC, @DESC+4, (R0)	
		6E	0130	C8	3C	00249	MOVZWL	DESC, R0	
			50	C0	0024E	ADDL2	R0, PTR		
			6E	D5	00251	21\$: TSTL	PTR		0558
			02	15	00253	BLEQ	22\$		
			6E	D7	00255	DECL	PTR		
			5E	DD	00257	22\$: PUSHL	SP		0560
			67	DD	00259	PUSHL	(R7)		
		0178	C8	9F	0025B	PUSHAB	P.AFX		
		0164	C8	9F	0025F	PUSHAB	P.AFV		
		69	04	FB	00263	CALLS	#4, EXCH\$UTIL_FAO_BUFFER		
		6A	50	DD	00266	PUSHL	R0		
			01	FB	00268	CALLS	#1, LIB\$PUT_OUTPUT		
		0C	A6	DD	0026B	PUSHL	12(R6)		0562
		7E	0B	A6	9A	0026E	MOVZBL	11(R6), -(SP)	
			0C	A6	DD	00272	PUSHL	12(R6)	
		7E	0B	A6	9A	00275	MOVZBL	11(R6), -(SP)	
			01B8	C8	9F	00279	PUSHAB	P.AGB	
			01A0	C8	9F	0027D	PUSHAB	P.AFZ	
		69	06	FB	00281	CALLS	#6, EXCH\$UTIL_FAO_BUFFER		



6A		50	DD	00284	PUSHL	R0		
		01	FB	00286	CALLS	#1, LIB\$PUT_OUTPUT		
	04	A6	DD	00289	PUSHL	4(R6)		0564
7E	03	A6	9A	0028C	MOVZBL	3(R6), -(SP)		
	04	A6	DD	00290	PUSHL	4(R6)		
7E	03	A6	9A	00293	MOVZBL	3(R6), -(SP)		
	01F8	C8	9F	00297	PUSHAB	P.AGF		
	01E0	C8	9F	0029B	PUSHAB	P.AGD		
69		06	FB	0029F	CALLS	#6, EXCH\$UTIL_FAO_BUFFER		
		50	DD	002A2	PUSHL	R0		
6A		01	FB	002A4	CALLS	#1, LIB\$PUT_OUTPUT		
	14	A6	9F	002A7	PUSHAB	20(R6)		0565
	023C	C8	9F	002AA	PUSHAB	P.AGJ		
	0228	C8	9F	002AE	PUSHAB	P.AGH		
69		03	FB	002B2	CALLS	#3, EXCH\$UTIL_FAO_BUFFER		
		50	DD	002B5	PUSHL	R0		
6A		01	FB	002B7	CALLS	#1, LIB\$PUT_OUTPUT		
	54	A6	DD	002BA	PUSHL	84(R6)		0572
7E	3D	A6	9A	002BD	MOVZBL	61(R6), -(SP)		
	50	A6	DD	002C1	PUSHL	80(R6)		
7E	3C	A6	9A	002C4	MOVZBL	60(R6), -(SP)		
	4C	A6	DD	002C8	PUSHL	76(R6)		
7E	3B	A6	9A	002CB	MOVZBL	59(R6), -(SP)		
	48	A6	DD	002CF	PUSHL	72(R6)		
7E	3A	A6	9A	002D2	MOVZBL	58(R6), -(SP)		
	44	A6	DD	002D6	PUSHL	68(R6)		
7E	39	A6	9A	002D9	MOVZBL	57(R6), -(SP)		
	40	A6	DD	002DD	PUSHL	64(R6)		
7E	38	A6	9A	002E0	MOVZBL	56(R6), -(SP)		
	0270	C8	9F	002E4	PUSHAB	P.AGL		
69		0D	FB	002E8	CALLS	#13, EXCH\$UTIL_FAO_BUFFER		
		50	DD	002EB	PUSHL	R0		
6A		01	FB	002ED	CALLS	#1, LIB\$PUT_OUTPUT		
	10	A6	DD	002F0	PUSHL	16(R6)		0573
	02C8	C8	9F	002F3	PUSHAB	P.AGP		
	02B4	C8	9F	002F7	PUSHAB	P.AGN		
69		03	FB	002FB	CALLS	#3, EXCH\$UTIL_FAO_BUFFER		
		50	DD	002FE	PUSHL	R0		
6A		01	FB	00300	CALLS	#1, LIB\$PUT_OUTPUT		
	10	A6	D5	00303	TSTL	16(R6)		0574
		08	13	00306	BEQL	23\$		
	10	A6	DD	00308	PUSHL	16(R6)		0575
FCF0	CF	01	FB	0030B	CALLS	#1, EXCH\$DBG_NAM_DUMP		
	50	01	D0	00310	MOVL	#1, R0		0577
		04	00	00313	RET			0578

; Routine Size: 788 bytes, Routine Base: EXCH\$DBG\_CODE + 0629



```

: 491 0579 1 GLOBAL ROUTINE exch$dbg_utl_print_devchar (dev : $bblock) = %SBTTL 'exch$dbg_utl_print_devchar'
: 492 0580 2 BEGIN
: 493 0581 2
: 494 0582 2 $dbgtrc_prefix ('dbg_utl_print_devchar> ');
: 495 0583 2 $trace_print_fao ('characteristics longword !XL', .dev [0,0,32,0]);
: 496 0584 2
: 497 0585 2 IF .dev [dev$V_rec]
: 498 0586 2 THEN
: 499 0587 2 $print_lit (' rec - device is record oriented');
: 500 0588 2 IF .dev [dev$V_ccl]
: 501 0589 2 THEN
: 502 0590 2 $print_lit (' ccl - carriage control device');
: 503 0591 2 IF .dev [dev$V_trm]
: 504 0592 2 THEN
: 505 0593 2 $print_lit (' trm - device is a terminal');
: 506 0594 2 IF .dev [dev$V_dir]
: 507 0595 2 THEN
: 508 0596 2 $print_lit (' dir - device is directory structured');
: 509 0597 2 IF .dev [dev$V_sdi]
: 510 0598 2 THEN
: 511 0599 2 $print_lit (' sdi - device is single directory structured');
: 512 0600 2 IF .dev [dev$V_sqd]
: 513 0601 2 THEN
: 514 0602 2 $print_lit (' sqd - sequential block-oriented device (i.e., magtape)');
: 515 0603 2 IF .dev [dev$V_spl]
: 516 0604 2 THEN
: 517 0605 2 $print_lit (' spl - device being spooled');
: 518 0606 2 IF .dev [dev$V_opr]
: 519 0607 2 THEN
: 520 0608 2 $print_lit (' opr - device is an operator');
: 521 0609 2 IF .dev [dev$V_rct]
: 522 0610 2 THEN
: 523 0611 2 $print_lit (' rct - disk contains rct (dec standard 166 disk)');
: 524 0612 2 IF .dev [dev$V_net]
: 525 0613 2 THEN
: 526 0614 2 $print_lit (' net - network device');
: 527 0615 2 IF .dev [dev$V_fod]
: 528 0616 2 THEN
: 529 0617 2 $print_lit (' fod - files-oriented device (i.e., disk and mt)');
: 530 0618 2 IF .dev [dev$V_dua]
: 531 0619 2 THEN
: 532 0620 2 $print_lit (' dua - device is dual ported');
: 533 0621 2 IF .dev [dev$V_shr]
: 534 0622 2 THEN
: 535 0623 2 $print_lit (' shr - device shareable');
: 536 0624 2 IF .dev [dev$V_gen]
: 537 0625 2 THEN
: 538 0626 2 $print_lit (' gen - device is a generic device');
: 539 0627 2 IF .dev [dev$V_avl]
: 540 0628 2 THEN
: 541 0629 2 $print_lit (' avl - device available for use');
: 542 0630 2 IF .dev [dev$V_mnt]
: 543 0631 2 THEN
: 544 0632 2 $print_lit (' mnt - device is mounted');
: 545 0633 2 IF .dev [dev$V_mbx]
: 546 0634 2 THEN
: 547 0635 2 $print_lit (' mbx - device is a mailbox');
```



```

: 548      0636 2      IF .dev [dev$v_dmt]
: 549      0637 2      THEN
: 550      0638 2          $print_lit (' dmt - device marked for dismount');
: 551      0639 2      IF .dev [dev$v_elg]
: 552      0640 2      THEN
: 553      0641 2          $print_lit (' elg - device has error logging enabled');
: 554      0642 2      IF .dev [dev$v_all]
: 555      0643 2      THEN
: 556      0644 2          $print_lit (' all - device is allocated');
: 557      0645 2      IF .dev [dev$v_for]
: 558      0646 2      THEN
: 559      0647 2          $print_lit (' for - device is mounted foreign (i.e., non-file structured)');
: 560      0648 2      IF .dev [dev$v_sw]
: 561      0649 2      THEN
: 562      0650 2          $print_lit (' swl - device is software write locked');
: 563      0651 2      IF .dev [dev$v_idv]
: 564      0652 2      THEN
: 565      0653 2          $print_lit (' idv - device capable of providing input');
: 566      0654 2      IF .dev [dev$v_odv]
: 567      0655 2      THEN
: 568      0656 2          $print_lit (' odv - device capable of providing output');
: 569      0657 2      IF .dev [dev$v_rnd]
: 570      0658 2      THEN
: 571      0659 2          $print_lit (' rnd - device allows random access');
: 572      0660 2      IF .dev [dev$v_rtm]
: 573      0661 2      THEN
: 574      0662 2          $print_lit (' rtm - device is realtime in nature');
: 575      0663 2      IF .dev [dev$v_rck]
: 576      0664 2      THEN
: 577      0665 2          $print_lit (' rck - device has read checking enabled');
: 578      0666 2      IF .dev [dev$v_wck]
: 579      0667 2      THEN
: 580      0668 2          $print_lit (' wck - device has write checking enabled');
: 581      0669 2
: 582      0670 2      RETURN 1;
: 583      0671 1      END;
```

.PSECT EXCH\$DBG\_PLIT,NOWRT,2

```

69 20 65 63 69 76 65 64 20 2D 20 63 65 72 09 00BDC P.AGS: .ASCII <9>\rec - device is record oriented\
74 6E 65 69 72 6F 20 64 72 6F 63 65 72 20 73 00BEB
64 65 00BFA
010E0020 00BFC P.AGR: .LONG 17694752
00000000' 00C00 .ADDRESS P.AGS
65 67 61 69 72 72 61 63 20 2D 20 6C 63 63 09 00C04 P.AGU: .ASCII <9>\ccl - carriage control device\<0>
65 63 69 76 65 64 20 6C 6F 72 74 6E 6F 63 20 00C13
00 00C22
00 00C23
010E001E 00C24 P.AGT: .ASCII <0>
00000000' 00C28 .LONG 17694750
00000000' 00C28 .ADDRESS P.AGU
69 20 65 63 69 76 65 64 20 2D 20 6D 72 74 09 00C2C P.AGW: .ASCII <9>\trm - device is a terminal\<0>
00 6C 61 6E 69 6D 72 65 74 20 61 20 73 00C3B
010E001B 00C48 P.AGV: .LONG 17694747
00000000' 00C4C .ADDRESS P.AGW
69 20 65 63 69 76 65 64 20 2D 20 72 69 64 09 00C50 P.AGY: .ASCII <9>\dir - device is directory structur\

```



```
72 74 73 20 79 72 6F 74 63 65 72 69 64 20 73 00C5F
72 75 74 63 75 00C6E
00 00 00 64 65 00C73
010E0025 00C78 P.AGX: .ASCII \ed\<0><0><0>
00000000 00C7C .LONG 17694757
00000000 00C80 P.AHA: .ADDRESS P.AGY
00000000 00C8F .ASCII <9>\sdi - device is single directory s\
00000000 00C9E
00000000 00CA3 .ASCII \structured\
010E002C 00CAC P.AGZ: .LONG 17694764
00000000 00CB0 .ADDRESS P.AHA
00000000 00CB4 P.AHC: .ASCII <9>\sqd - sequential block-oriented de\
00000000 00CC3
00000000 00CD2
00000000 00CD7 .ASCII \vice (i.e., magtape)\<0>
00000000 00CE6
010E0037 00CEC P.AHB: .LONG 17694775
00000000 00CF0 .ADDRESS P.AHC
00000000 00CF4 P.AHE: .ASCII <9>\spl - device being spooled\<0>
00000000 00D03
010E001B 00D10 P.AHD: .LONG 17694747
00000000 00D14 .ADDRESS P.AHE
00000000 00D18 P.AHG: .ASCII <9>\opr - device is an operator\
00000000 00D27
010E001C 00D34 P.AHF: .LONG 17694748
00000000 00D38 .ADDRESS P.AHG
00000000 00D3C P.AHI: .ASCII <9>\rct - disk contains rct (dec stand\
00000000 00D4B
00000000 00D5A
00000000 00D5F .ASCII \ard 166 disk)\
010E0030 00D6C P.AHH: .LONG 17694768
00000000 00D70 .ADDRESS P.AHI
00000000 00D74 P.AHK: .ASCII <9>\net - network device\<0><0><0>
00000000 00D83
010E0015 00D8C P.AHJ: .LONG 17694741
00000000 00D90 .ADDRESS P.AHK
00000000 00D94 P.AHM: .ASCII <9>\fod - files-oriented device (i.e.,\
00000000 00DA3
00000000 00DB2
00000000 00DB7 .ASCII \ disk and mt)\
010E0030 00DC4 F.AHL: .LONG 17694768
00000000 00DC8 .ADDRESS P.AHM
00000000 00DCC P.AHO: .ASCII <9>\dua - device is dual ported\
00000000 00DD6
010E001C 00DE8 P.AHN: .LONG 17694748
00000000 00DEC .ADDRESS P.AHO
00000000 00DF0 P.AHQ: .ASCII <9>\shr - device shareable\<0>
00000000 00DFF
010E0017 00E08 P.AHP: .LONG 17694743
00000000 00E0C .ADDRESS P.AHQ
00000000 00E10 P.AHS: .ASCII <9>\gen - device is a generic device\<0>
00000000 00E1F
00000000 00E2E
00000000 00E32 .ASCII <0><0>
010E0021 00E34 P.AHR: .LONG 17694753
00000000 00E38 .ADDRESS P.AHS
00000000 00E3C P.AHU: .ASCII <9>\avl - device available for use\<0>
```



[illegible]



```
010E0022 01024 P.AIN: .LONG 17694754
00000000' 01028 .ADDRESS P.AIO
69 20 65 63 69 76 65 64 20 2D 20 6D 74 72 09 0102C P.AIQ: .ASCII <9>\rtm - device is realtime in nature\
6E 20 6E 69 20 65 6D 69 74 6C 61 65 72 20 73 0103B
65 72 75 74 61 0104A
00 0104F
010E0023 01050 P.AIP: .ASCII <0>
00000000' 01054 .LONG 17694755
68 20 65 63 69 76 65 64 20 2D 20 6B 63 72 09 01058 P.AIS: .ADDRESS P.AIQ
6E 69 6B 63 65 68 63 20 64 61 65 72 20 73 61 01054 P.AIS: .ASCII <9>\rck - device has read checking ena\
61 6E 65 20 67 01067
00 64 65 6C 62 01076
010E0027 01080 P.AIR: .ASCII \bled\<0>
00000000' 01084 .LONG 17694759
68 20 65 63 69 76 65 64 20 2D 20 6B 63 77 09 01088 P.AIU: .ADDRESS P.AIS
69 6B 63 65 68 63 20 65 74 69 72 77 20 73 61 01088 P.AIU: .ASCII <9>\wck - device has write checking en\
6E 65 20 67 6E 01097
64 65 6C 62 61 010A6
010E0028 010B0 P.AIT: .ASCII \abled\
00000000' 010B4 .LONG 17694760
. ADDRESS P.AIU
```

.PSECT EXCH\$DBG\_CODE,NOWRT,2

```
0004 00000
52 00000000G 00 9E 00002 .ENTRY EXCH$DBG_UTL_PRINT_DEVCHAR, Save R2
07 04 AC 0000' 01 FB 00009 MOVAB LIB$PUT_OUTPUT, R2
0000' 01 FB 00011 BLBC DEV, 1$
07 04 AC 0000' 01 E1 00014 1$: CALLS #1, LIB$PUT_OUTPUT
0000' 01 FB 00019 BBC #1, DEV, 2$
07 04 AC 0000' 02 E1 00020 2$: PUSHAB P.AGR
0000' 02 E1 00025 CALLS #1, LIB$PUT_OUTPUT
07 04 AC 0000' 03 E1 0002C 3$: BBC #2, DEV, 3$
0000' 03 E1 00031 PUSHAB P.AGT
07 04 AC 0000' 04 E1 00038 4$: CALLS #1, LIB$PUT_OUTPUT
0000' 04 E1 0003D BBC #3, DEV, 4$
07 04 AC 0000' 05 E1 00041 5$: PUSHAB P.AGV
0000' 05 E1 00049 CALLS #1, LIB$PUT_OUTPUT
07 04 AC 0000' 06 E1 0004D 6$: BBC #4, DEV, 5$
0000' 06 E1 00055 PUSHAB P.AGX
07 04 AC 0000' 07 18 0005F 7$: CALLS #1, LIB$PUT_OUTPUT
0000' 07 18 00061 TSTB DEV
07 04 AC 0000' 01 FB 00065 BGEQ 8$
0000' 01 FB 00068 PUSHAB P.AGZ
07 04 AC 0000' 01 FB 00070 CALLS #1, LIB$PUT_OUTPUT
0000' 01 FB 00073 BBC #5, DEV, 6$
07 05 AC 0000' 05 E1 00078 9$: PUSHAB P.AHZ
0000' 05 E1 00078 CALLS #1, LIB$PUT_OUTPUT
0000' 05 E1 00078 BLBC DEV+1, 9$
0000' 05 E1 00078 PUSHAB P.AHF
0000' 05 E1 00078 CALLS #1, LIB$PUT_OUTPUT
0000' 05 E1 00078 BBC #6, DEV, 7$
0000' 05 E1 00078 PUSHAB P.AHD
0000' 05 E1 00078 CALLS #1, LIB$PUT_OUTPUT
0000' 05 E1 00078 BBC #5, DEV+1, TOS
0000' 05 E1 00078 PUSHAB P.AHJ
```



07	05	62	01	FB	0007C	CALLS	#1, LIB\$PUT_OUTPUT	0615
		AC	06	E1	0007F	BBC	#6, DEV+1, T1\$	0617
			CF	9F	00084	PUSHAB	P.AHL	
		62	01	FB	00088	CALLS	#1, LIB\$PUT_OUTPUT	0618
			05	AC	95 0008B	TSTB	DEV+1	
			07	18	0008E	BGEQ	12\$	0620
			CF	9F	00090	PUSHAB	P.AHN	
		62	01	FB	00094	CALLS	#1, LIB\$PUT_OUTPUT	0621
		07	06	AC	E9 00097	BLBC	DEV+2, 13\$	0623
			CF	9F	0009B	PUSHAB	P.AHP	
07	06	62	01	FB	0009F	CALLS	#1, LIB\$PUT_OUTPUT	0624
		AC	01	E1	000A2	BBC	#1, DEV+2, T4\$	0626
			CF	9F	000A7	PUSHAB	P.AHR	
07	06	62	01	FB	000AB	CALLS	#1, LIB\$PUT_OUTPUT	0627
		AC	02	E1	000AE	BBC	#2, DEV+2, T5\$	0629
			CF	9F	000B3	PUSHAB	P.AHT	
07	06	62	01	FB	000B7	CALLS	#1, LIB\$PUT_OUTPUT	0630
		AC	03	E1	000BA	BBC	#3, DEV+2, T6\$	0632
			CF	9F	000BF	PUSHAB	P.AHV	
07	06	62	01	FB	000C3	CALLS	#1, LIB\$PUT_OUTPUT	0633
		AC	04	E1	000C6	BBC	#4, DEV+2, T7\$	0635
			CF	9F	000CB	PUSHAB	P.AHX	
07	06	62	01	FB	000CF	CALLS	#1, LIB\$PUT_OUTPUT	0636
		AC	05	E1	000D2	BBC	#5, DEV+2, T8\$	0638
			CF	9F	000D7	PUSHAB	P.AHZ	
07	06	62	01	FB	000DB	CALLS	#1, LIB\$PUT_OUTPUT	0639
		AC	06	E1	000DE	BBC	#6, DEV+2, T9\$	0641
			CF	9F	000E3	PUSHAB	P.AIB	
		62	01	FB	000E7	CALLS	#1, LIB\$PUT_OUTPUT	0642
			06	AC	95 000EA	TSTB	DEV+2	
			07	18	000ED	BGEQ	20\$	0644
			CF	9F	000EF	PUSHAB	P.AID	
		62	01	FB	000F3	CALLS	#1, LIB\$PUT_OUTPUT	0645
		07	07	AC	E9 000F6	BLBC	DEV+3, 21\$	0647
			CF	9F	000FA	PUSHAB	P.AIF	
07	07	62	01	FB	000FE	CALLS	#1, LIB\$PUT_OUTPUT	0648
		AC	01	E1	00101	BBC	#1, DEV+3, 22\$	0650
			CF	9F	00106	PUSHAB	P.AIH	
07	07	62	01	FB	0010A	CALLS	#1, LIB\$PUT_OUTPUT	0651
		AC	02	E1	0010D	BBC	#2, DEV+3, 23\$	0653
			CF	9F	00112	PUSHAB	P.AIJ	
07	07	62	01	FB	00116	CALLS	#1, LIB\$PUT_OUTPUT	0654
		AC	03	E1	00119	BBC	#3, DEV+3, 24\$	0656
			CF	9F	0011E	PUSHAB	P.AIL	
07	07	62	01	FB	00122	CALLS	#1, LIB\$PUT_OUTPUT	0657
		AC	04	E1	00125	BBC	#4, DEV+3, 25\$	0659
			CF	9F	0012A	PUSHAB	P.AIN	
07	07	62	01	FB	0012E	CALLS	#1, LIB\$PUT_OUTPUT	0660
		AC	05	E1	00131	BBC	#5, DEV+3, 26\$	0662
			CF	9F	00136	PUSHAB	P.AIP	
07	07	62	01	FB	0013A	CALLS	#1, LIB\$PUT_OUTPUT	0663
		AC	06	E1	0013D	BBC	#6, DEV+3, 27\$	0665
			CF	9F	00142	PUSHAB	P.AIR	
		62	01	FB	00146	CALLS	#1, LIB\$PUT_OUTPUT	0666
			07	AC	95 00149	TSTB	DEV+3	
			07	18	0014C	BGEQ	28\$	0668
			CF	9F	0014E	PUSHAB	P.AIT	



EXCH\$DBG  
V04-000

Conditionally assembled debugging routines  
exch\$dbg\_u~~tl~~\_print\_devchar

M 7  
16-Sep-1984 00:45:26  
14-Sep-1984 12:29:02

VAX-11 Bliss-32 V4.0-742  
[EXCHNG.SRC]EXCDBG.B32;1

Page 39  
(9)

62  
50

01 FB 00152  
01 D0 00155  
04 00158

28\$:

CALLS #1, LIB\$PUT\_OUTPUT  
MOVL #1, R0  
RET

: 0670  
: 0671

; Routine Size: 345 bytes,      Routine Base: EXCH\$DBG\_CODE + 093D



```

: 585      0672 1 END
: 586      0673 0 ELUDOM

```

```

      .EXTRN LIB$SIGNAL, LIB$STOP

```

# PSECT SUMMARY

Name	Bytes	Attributes					
EXCH\$DBG_PLIT	4280	NOVEC,NOWRT,	RD	EXE,NOSHR,	LCL,REL,	CON,NOPIC,ALIGN(2)	
EXCH\$DBG_CODE	2710	NOVEC,NOWRT,	RD	EXE,NOSHR,	LCL,REL,	CON,NOPIC,ALIGN(2)	
EXCH\$DBG_OWN	180	NOVEC, WRT,	RD	NOEXE,NOSHR,	LCL,REL,	CON,NOPIC,ALIGN(2)	

# Library Statistics

File	-----		Symbols		-----		Pages	Processing
	Total		Loaded	Percent				
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619		106	0		1000		00:01.9
_\$255\$DUA28:[EXCHNG.OBJ]EXCLIB.L32;1	1151		56	4		79		00:01.4

# COMMAND QUALIFIERS

```

:      BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:EXCDBG/OBJ=OBJ$:EXCDBG MSRC$:EXCDBG/UPDATE=(ENH$:EXCDBG)

```

```

: Size:      2710 code + 4460 data bytes
: Run Time:   00:53.1
: Elapsed Time: 02:55.9
: Lines/CPU Min: 760
: Lexemes/CPU-Min: 26418
: Memory Used: 348 pages
: Compilation Complete

```



0160 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

